

WORKINGPAPER

**Omzetting van complexe beurskoersinformatie
in een relationele databankstructuur.**

**De Beurs van Brussel, 1830 - 2002:
van een theoretisch ontwerp naar een geslaagde
implementatie.**

**Nele Dexters
Dept. Wiskunde-Informatica
UIA**

<u>INHOUDSTAFEL</u>	2
Inleiding	4
Deel I De Databank	7
1. Inleiding	8
2. Algemene Principes	8
3. Invoermogelijkheden	9
4. Uitvoermogelijkheden	10
5. Wat zit er nu concreet in de databank?	10
5.1. Archivenumber	14
5.2. Archiver	15
5.3. Bank	16
5.4. Bookkeepingitem	17
5.5. Category	18
5.6. Codetype	18
5.7. Corporation	20
5.8. Corporationaction	20
5.9. Corporationset	21
5.10. Corporation_adminlocation	21
5.11. Corporation_bank	22
5.12. Corpoartion_bookkeeping	22
5.13. Corporation_bookkeepingitem	23
5.14. Corporation_capitalchanges	23
5.15. Corporation_evolution	24
5.16. Corporation_information	24
5.17. Corporation_location	25
5.18. Corporation_name	25
5.19. Corporation_resultitems	26
5.20. Corporation_results	26
5.21. Corporation_sociallocation	27
5.22. Corporation_stockholders	27
5.23. Corporation_stocks	28
5.24. Currency	28
5.25. Dividendtype	29
5.26. Document	29
5.27. Documenttype	30
5.28. Document_san	31
5.29. Document_title	32
5.30. Evolutionrole	33
5.31. Job	33
5.32. Notation	34
5.33. Notation_price	34
5.34. Notation_tradedquantity	36
5.35. Person	36
5.36. Person_function	37
5.37. Person_location	37
5.38. Quantitytype	38

5.39.	Resultitems	38
5.40.	Sector	39
5.41.	Sector_child	41
5.42.	Sector_name	41
5.43.	Sector_next	42
5.44.	Sessionpart	43
5.45.	Sharetype	44
5.46.	Stock	44
5.47.	Stockexchange	45
5.48.	Stockexchange_child	46
5.49.	Stockexchange_name	47
5.50.	Stocktype	48
5.51.	Stock_category	49
5.52.	Stock_codetype	49
5.53.	Stock_corporation	50
5.54.	Stock_dividend	51
5.55.	Stock_emission	52
5.56.	Stock_name	52
5.57.	Stock_nominal	53
5.58.	Stock_quantity	54
5.59.	Stock_splits	54
5.60.	Stock_type	56
5.61.	Subarchivenummer	56
5.62.	Valuta	60
6.	Logische samenhang van de tabellen	60
6.1.	De Aandeelmodule	61
6.2.	De Bedrijfsmodule	63
6.3.	De Documentmodule	65
Deel II Modules		66
1.	Inleiding	67
2.	De Aandeelmodule	68
I.	Het Linkerblok	70
II.	Het Rechterblok	71
II.a.	Sectortitel	71
II.b.	Aandeel- of Obligatietitel	74
III.	De Statusbalk	74
IV.	Het Aanmaken van Aandelen	75
3.	De Bedrijfsmodule	80
4.	De Outputmodule	84
5.	De Documentmodule	87
6.	Makestockinfo	88
7.	MakeMatrix	91
8.	De Logboeken	92
8.1.	Logboek Aandelen	92
8.2.	Logboek Obligaties	94
8.3.	Logboek Stock Splits	95

Inleiding

Om wetenschappelijk onderzoek naar het beurswezen van België mogelijk te maken, werd er in 1995 door de Beurs van Brussel met de Universiteit van Antwerpen een overeenkomst gesloten waarbij de archieven van de Beurs van Brussel ter beschikking werden gesteld van de universiteit. De Beurs van Brussel stond haar authentieke koerslijsten af aan de Universiteit Antwerpen die zich op haar beurt engageerde om deze lijsten in goede staat te bewaren en door wetenschappelijk onderzoek te ontsluiten.

Als gevolg hiervan werd er een grootscheeps onderzoeksproject (GOA) "**Catalogisering en digitalisering van het archief van de Beurs van Brussel**" (1998-2002) opgesteld, met als bedoeling het archief van de Beurs van Brussel toegankelijk te maken door de bedrijfshistorische informatie te inventariseren en de koersinformatie te digitaliseren. Hiervoor diende er de nodige software ontwikkeld te worden. Het achterliggende doel van het project is natuurlijk praktisch onderzoek te kunnen verrichten op deze rijke, nooit eerder onderzochte gegevensbron.

Het project is opgezet onder leiding van een interdisciplinair team van de toenmalige drie Antwerpse universitaire instellingen, RUCA, UFSIA en UIA (faculteiten "Toegepaste Economische Wetenschappen", "Geschiedenis" en "Wiskunde-Informatica"). De stuurgroep wordt gevormd door prof. dr. L. Cuyvers, de voorzitter, en de professoren J. Annaert, M. De Ceuster, G. Devos, H. Houtman - De Smedt en J. Paredaens. Zij kunnen beroep doen op de medewerking van drie full-time medewerkers, een financieel econoom, een historicus en een informaticus, en dit om dit alles in een langetermijnvisie in te passen. Hun werkplek is omgedoopt tot "SCOB" (sinds 1999), de afkorting voor "**StudieCentrum voor Onderneming en Beurs**", en heeft haar lokaal A. 142 op de Middelheimcampus in het Koloniale Gebouw. Daar liggen alle archieven, knipsels, boeken, koerslijsten en kranten die er gebruikt worden voor de invoer in de Oracle-databank, die ook daar terug te vinden is, net als de SUN-server en de webserver waarop de website van het SCOB draait (<http://www.scob.be>) en een tiental computers (waaronder een laptop) voor de data-invoer.

Ondertussen heeft het SCOB, naast alle gegevens van de Beurs van Brussel (ook PDF-files met alle actuele info inzake het beurswezen, die op driemaandelijks basis op CD gebrand worden), ook nog andere collecties

verworven, ondermeer van "de Kredietbank", "de Beurs van Antwerpen", "de Beurs van Gent", "Echo de la Bourse" en "de Commissie voor het Bank- en Financiewezen".

Er is door de projectgroep een databank ontwikkeld die toelaat om voor het eerst in België te beschikken over computerleesbare historische reeksen van beurskoersen en dividendgegevens zodat het voor financiële economen mogelijk wordt om te onderzoeken hoe aandelenkoersen statistisch verdeeld zijn en of deze verdeling stabiel blijft in de tijd. Dergelijke lange historische reeksen zijn onontbeerlijk datamateriaal dat zeldzaam is op wereldschaal en voor België onbestaande was totdat dit project startte. Voor de allereerste keer is het rijke bedrijfs- en beursverleden van België volledig geïnventariseerd in één centrale databank waaruit talloze draden van onderzoek mogelijk zullen zijn. De universiteit Antwerpen beschikt bijgevolg op dit moment over een vrijwel **uniek en bijzonder uitgebreid archief van beurs- en bedrijfsgegevens**.

Over de brainstorm voor de ontwikkeling van deze databank is er reeds een working paper verschenen, namelijk working paper **98/11**, met als titel "**Ontwerp van een databank m.b.t. het Archief van de Beurs van Brussel**". Sindsdien werden de daarin beschreven verwachtingen ook geconcretiseerd. **Het uiteindelijke resultaat wordt in deze paper beschreven.**

In deze working paper wordt er in het bijzonder gekeken naar de informaticazijde van het project. Het belangrijkste aspect hierbij is het ontwerp van de databank. Eens deze bestaat, moeten de gegevens er op een gebruiksvriendelijke en vooral snelle manier in ingevoerd kunnen worden en er natuurlijk ook weer uit opgevraagd kunnen worden voor onderzoeksdoeleinden. De databank moet niet enkel beurs- en bedrijfsgegevens kunnen opnemen van de Beurs van Brussel, maar ook van andere beurzen uit binnen- en buitenland, zoals de Beurs van Antwerpen en Gent. Daarnaast is er ruimte voorzien om aan de diverse bedrijven allerlei belangrijke documenten en gegevens te koppelen die het mogelijk moeten maken om het geschiedkundig en economisch onderzoek te verrijken.

Over het ontwerp van de databank is er zoals reeds eerder vermeld reeds (in 1998) een working paper verschenen. Deze ging enkel over het ontwerp, dus het "geraamte" van de databank. Het is geenszins de bedoeling om deze nog eens opnieuw te maken, eerder **om de databank**

die er ondertussen **gerealiseerd is** uitgaande van deze “kapstok” en die optimaal gebruikt wordt, te **toetsen** aan de **voorstellen** en het **ontwerp** uit de working paper.

Verder wordt er ook ruimschoots aandacht besteed aan de **modules** die er geschreven zijn om op een eenvoudige manier gegevens in te voeren in de databank, maar ook gegevens op een overzichtelijke manier terug op te vragen.

Deze workingpaper is opgedeeld in **twee** grote delen.

Deel I gaat over de bovengenoemde SCOB-databank en **Deel II** behandelt de verschillende modules die er geschreven zijn “op” deze databank en die dagelijks actief gebruikt worden voor de verwerking van data. Deel II handelt bijgevolg over *hoe* we nu concreet werken met de gecreëerde tabellen.

Deel I

De Databank

1. Inleiding

De vorige working paper die er verschenen is over de databank is ingedeeld in drie grote delen.

Het eerste deel handelde over "Het Model", het tweede deel over "De Structuur van het Archief van de Beurs van Brussel" en het derde deel en tevens het grootste was de eigenlijke paper die de uitgebreide structuur van de databank beschreef. De nadruk lag bijgevolg op het derde deel, de eigenlijke databankstructuur.

Deze wordt nu kritisch onder de loep genomen en besproken.

2. Algemene principes

De databank die het SCOB gebruikt is een Oracle databank, die draait op een SUN-server en die vanuit meerdere terminals ingevuld en geconsulteerd kan worden. Op dit moment is er versie 8.0.5 van de Oracle databank geïnstalleerd en actief. Upgrades kunnen van het net gedownload worden, maar we doet dit niet. Ook upgraden naar Oracle 9 gebeurt er niet. Dit zou een upgraden van de SUN vereisen.

We hebben er bewust voor gekozen om te blijven werken met de versie die er nu draait omdat in deze versie alle functionaliteiten zitten die we vereisen. Upgraden zou ons een heleboel extra features opleveren die toch niet gebruikt worden plus de eventuele kans op bugs en andere fouten die een upgrade met zich meebrengt.



Het meest karakteristieke aan de SCOB databank is dat deze werkt met **temporeel veranderlijke data**. Om dit te realiseren krijgen de data een begin- en einddatum. Men moet dit interpreteren als de "geldigheidsperiode" van de betrokken data.

In de SCOB-databank vinden we ondertussen alle aandelen terug die ooit op de Brusselse Beurs noteerden, samen met de bedrijven die deze aandelen uitgaven. Van de bedrijven worden diverse gegevens bijgehouden (datum van de oprichting, maatschappelijke zetel,...) net als van de aandelen (maandelijkse koersen, dividenden, hoeveelheden,...) en dit voor de periode 1832-2002 (nu). Voor obligaties is er een gelijkaardige opzet voorzien. Ook intresten zijn verwerkt.

De documenten die door de Beurs van Brussel aan de UA zijn geschonken zijn volledig geïnventariseerd in een Word-document dat binnenkort

gepubliceerd zal worden. Het is de bedoeling om deze inventaris ook in de databank in te voeren en te verwerken via de "Documentmodule" (zie Deel II, Modules), zodat het als output beschikbaar gesteld kan worden en het de raadpleging van de archieven zeker versnelt en vergemakkelijkt.

3. Invoermogelijkheden

De Oracle databank laat toe om de meest uiteenlopende data **op te slaan** in tabellen die door de informatici die meewerk(t)en aan het project zijn opgesteld. De invoer die er op het SCOB gebeurt, gebeurt **nooit rechtstreeks** in de Oracle databank zelf, maar wel via **tussenmodules**. In feite zijn er **drie** mogelijkheden om de databank "aan te spreken".

1. Er zijn **specifieke invoermodules** geschreven (zie **Deel II, Modules**) voor aandelen en obligaties en voor bedrijfsinformatie. Invoeren via de invoermodules is de meest aangewezen vorm van data input omdat er hier een aantal "interne" controles ingebouwd kunnen worden en het zeer gebruiksvriendelijk is. Men heeft hiervoor immers geen "databankkennis" nodig. Het invoerproces wordt als het ware losgekoppeld van de eigenlijke databank. We vermelden hier de "Aandelenmodule" (deze module omvat diverse componenten voor koersen, hoeveelheden aandelen, dividenden, nominale waarde, verhandelde volumes en een onderdeel om nieuwe aandelen te creëren, en soortgelijke aspecten voor obligaties), de "Bedrijfsmodule" (deze module laat toe een bedrijf in te voeren en de hierbij horende informatie over aandeelhouders, boekhouding enz.) en de "Documentmodule" (deze module dient om het archief van het SCOB op te slaan en te raadplegen m.b.v. de databank). Deze worden in detail besproken in **Deel II, Modules**.
2. Er kan rechtstreeks ingevoerd worden via **Access**. Deze manier van invoeren situeert zich meer in de databanksfeer aangezien Access de mogelijkheid is die Microsoft aanbiedt om te werken met databanken. Er is een link gelegd tussen Oracle en Access waardoor het mogelijk is om in Access te werken terwijl de wijzigingen rechtstreeks in de Oracle databank worden doorgevoerd. Men kan via deze link gegevens rechtstreeks invoeren, consulteren en eventueel aanpassen of verwijderen. Alle tabellen in Oracle kunnen also opgeroepen worden in Access en gebruikt worden **alsof** men rechtstreeks op de databank werkt. De toegang tot de databank via

Access wordt **zo veel mogelijk beperkt** gehouden omdat de kans zeer groot is dat er invoer- en andere fouten gemaakt worden.

3. Het is ook mogelijk om rechtstreeks via **SQL** data toe te voegen of te veranderen. Deze manier om te data te bereiken wordt enkel gebruikt door de informaticamedewerkers van SCOB.

4. Uitvoermogelijkheden

Ook om informatie **uit** de Oracle databank **te halen** zijn er verschillende mogelijke strategieën.

1. Via **SQL-statements** kan men rechtstreeks informatie uit de tabellen opvragen. Dit kan gebeuren via de "Oracle Enterprise Manager / SQL Worksheet" en is uiterst gebruiksvriendelijk. Het vraagt wel de nodige voorkennis van de databank query taal SQL natuurlijk.
2. Er is ook een speciaal ontworpen module, de "**Grafische module**" of "**Outputmodule**" waar de gebruiker een aantal vaste SQL-statements kan kiezen en kan laten uitvoeren. Hij kan ook zelf zijn eigen SQL-statements invoeren. Er bestaat ook de mogelijkheid om de opgevraagde gegevens te visualiseren d.m.v. grafieken. Voor de indexberekening is er een aparte module geschreven, die geïntegreerd zit in "de Grafische Module" (zie **Deel II , Modules**).

5. Wat zit er nu concreet in de databank?

Op dit moment zitten er **62** tabellen in de databank, elk met hun specifieke functie. We zullen deze apart bespreken met hun voornaamste kenmerken.

Als we met SQL de Oracle-databank ondervragen naar zijn inhoud (zie **4.1.**), krijgen we als **inhoud** de volgende tabellen terug.

TABLE_NAME	TABLE_TYPE
ARCHIVENUMBER	TABLE
ARCHIVER	TABLE
BANK	TABLE
BOOKKEEPINGITEM	TABLE
CATEGORY	TABLE
CODETYPE	TABLE
CORPORATION	TABLE
CORPORATIONACTION	TABLE
CORPORATIONSET	TABLE
CORPORATION_ADMINLOCATION	TABLE
CORPORATION_BANK	TABLE
CORPORATION_BOOKKEEPING	TABLE
CORPORATION_BOOKKEEPINGITEM	TABLE
CORPORATION_CAPITALCHANGES	TABLE
CORPORATION_EVOLUTION	TABLE
CORPORATION_INFORMATION	TABLE
CORPORATION_LOCATION	TABLE
CORPORATION_NAME	TABLE
CORPORATION_RESULTITEMS	TABLE
CORPORATION_RESULTS	TABLE
CORPORATION_SOCIALLOCATION	TABLE
CORPORATION_STOCKHOLDERS	TABLE
CORPORATION_STOCKS	TABLE
CURRENCY	TABLE
DIVIDENDTYPE	TABLE
DOCUMENT	TABLE
DOCUMENTTYPE	TABLE
DOCUMENT_SAN	TABLE
DOCUMENT_TITLE	TABLE
EVOLUTIONROLE	TABLE
JOB	TABLE
NOTATION	TABLE
NOTATION_PRICE	TABLE
NOTATION_TRADEDQUANTITY	TABLE
PERSON	TABLE
PERSON_FUNCTION	TABLE
PERSON_LOCATION	TABLE
QUANTITYTYPE	TABLE
RESULTITEMS	TABLE
SECTOR	TABLE
SECTOR_CHILD	TABLE
SECTOR_NAME	TABLE
SECTOR_NEXT	TABLE
SESSIONPART	TABLE
SHARETYPE	TABLE
STOCK	TABLE
STOCKEXCHANGE	TABLE
STOCKEXCHANGE_CHILD	TABLE
STOCKEXCHANGE_NAME	TABLE
STOCKTYPE	TABLE
STOCK_CATEGORY	TABLE
STOCK_CODETYPE	TABLE
STOCK_CORPORATION	TABLE
STOCK_DIVIDEND	TABLE
STOCK_EMISSION	TABLE
STOCK_NAME	TABLE
STOCK_NOMINAL	TABLE
STOCK_QUANTITY	TABLE

```
STOCK_SPITS          TABLE
STOCK_TYPE           TABLE
SUBARCHIVENUMBER    TABLE
VALUTA               TABLE

62 rows selected.
```

Wat er onmiddellijk opvalt bij het bekijken van deze tabellen is dat sommigen bestaan uit één enkele naam, terwijl anderen opgebouwd zijn uit meerdere delen die met elkaar verbonden zijn d.m.v. het “_”-teken. De tabellen met een naam die bestaat uit één enkel woord noemen we *enkelvoudig*, de tabellen die bestaan uit meerdere woorden gescheiden door een “_” noemen we *samengesteld*. De samengestelde tabellen zijn de tabellen waarin de informatie opgeslagen is die uit de archieven van SCOB komt, de “data”. Zo geeft bijvoorbeeld de tabel `stock_type` per stock het type van die stock weer. Er moet natuurlijk ergens in de databank gespecificeerd zijn welke types een stock kan hebben, en daarvoor dient de tabel `stocktype`. De enkelvoudige tabellen zijn bijgevolg de tabellen die de structuren bepalen die er in de samengestelde tabellen gebruikt gaan worden. De concrete werking van deze koppeling wordt in detail beschreven bij de desbetreffende tabellen.

We zullen nu elk van de 62 tabellen **apart** bespreken. Tussen haken is terug te vinden in welke module (zie **Deel II, Modules** voor meer uitleg over de modules zelf) deze tabel hoofdzakelijk gebruikt gaat worden. Per tabel wordt er duidelijk weergegeven hoe deze in Oracle is opgevat. Eerst worden er in een kader de tabelspecificaties van de tabel waar het over gaat gegeven, en daarna worden deze in woorden uitgelegd, eventueel aangevuld met een of meerdere voorbeelden. Deze voorbeelden komen rechtstreeks uit de databank en zijn ook in een kader weergegeven, met als titel de naam van de tabel. Zij bestaan dus uit een selectie tupels uit de concrete invulling van de desbetreffende tabel zoals deze in de databank terug te vinden is.

We verklaren nu kort hoe de **tabelspecificaties** van Oracle, die er bij het begin van de bespreking van elke tabel terug te vinden zijn, geïnterpreteerd moeten worden. De informatie, die met SQL uit de databank verkregen is, is omkaderd en beschrijft hoe de tabel is opgebouwd. `Column Name` geeft de namen van kolommen (attributen) in de desbetreffende tabel. `Null?` geeft aan of deze informatie al dan niet verplicht ingevuld moet worden. Als er expliciet staat `NOT NULL` betekent dit dat wat er daar ingevuld moet worden *niet leeg* kan zijn, dus expliciet

vermeld "moet" worden. Als er niks staat, betekent dit dan ook dat er niks ingevuld *hoeft* te worden, doch *wel mag* (en natuurlijk ook liefst gebeurt, zodat we over zo veel mogelijk informatie kunnen beschikken). `TYPE` omschrijft van welke hoedanigheid de in te vullen informatie dient te zijn. Dit kan bijvoorbeeld `NUMBER` zijn, dus een getal, of `VARCHAR(200)`, een opeenvolging van maximaal 200 letters (karakters) of `DATE`, een datum. Gewoonlijk zijn deze typeringen wel duidelijk.



We hebben er hier bewust voor gekozen om de tabellen **in alfabetische volgorde** te doorlopen, zodat dit overzicht gebruikt kan worden als **naslagwerk**. Op deze manier kunnen de afzonderlijke tabellen en hun specificaties immers snel teruggevonden worden. Er wordt wél zo veel mogelijk vermeld welke tabellen "bij elkaar" horen en hoe dit gerealiseerd is. Dit alles wordt duidelijk geïllustreerd met concrete voorbeelden uit de databank. Zo heeft de lezer die één enkele tabel opzoekt, toch een beeld van de interne samenhang van deze tabellen met de andere tabellen. Wil men echter **een globaal beeld** van alle tabellen en de bijbehorende modules om zo een algemeen overzicht te krijgen van de werking van de SCOB-databank, dan kan men punt **6**. raadplegen. Hierin wordt er dieper ingegaan op **de logische samenhang** van de tabellen.



Wat de data (jaren, maanden en dagen) betreft die er in de SCOB-databank terug te vinden zijn, gelden de volgende afspraken. Er wordt zo veel mogelijk getracht de **exacte datum** weer te geven zoals deze in de bronnen terug te vinden is. Vaak wordt deze zelfs meermaals in verschillende bronnen gecheckt. Als we echter niet beschikken over een officiële startdatum wordt er standaard gewerkt met "1/1/1000". Als de "echte" einddatum niet gekend is, werken we met "31/12/3999".



In de SCOB-databank wordt er ook zo veel mogelijk gewerkt met de **originele benamingen** zoals deze terug te vinden zijn in de koersboeken. Deze zijn hoofdzakelijk in het Frans. Het SCOB heeft er bewust voor gekozen om zeer kort bij de originele documenten te blijven omdat dit voor de operators de herkenbaarheid verhoogt. Gewoonlijk staat er ter verduidelijking wel de Nederlandse vertaling bij tussen haakjes.

5.1. Archivenummer (Documentmodule)

Column Name	Null?	Type
-----	-----	----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(200)
ARCHIVER		NUMBER
PUBLISHER		NUMBER
STARTDATE		DATE
ENDDATE		DATE

De tabel *Archivenummer* beschrijft de archieven die er op dit moment in het bezit zijn van SCOB.

Elk archief (de volledige naam is terug te vinden in *NAME*) krijgt een uniek nummer (*ID*) waarmee het geïdentificeerd kan worden. Tevens wordt er bijgehouden wie het archief heeft bijgehouden (*ARCHIVER*) en wie de uitgever was (*PUBLISHER*). Om te weten wie er precies bedoeld wordt met de unieke identificatiesleutel *ARCHIVER*, moet men de tabel *Archiver* raadplegen (5.2.) Hieruit kan men dan de volledige naam van de archiefvormer, de instelling of persoon die het archief of de documenten heeft bijgehouden, halen. Met *PUBLISHER* wordt de persoon/onderneming/instelling bedoeld die het document in kwestie daadwerkelijk heeft uitgegeven of opgesteld of gepubliceerd. In het geval van de bedrijfsdossiers van de Beurs van Brussel gaat het dus om bedrijven. *PUBLISHER* verwijst dan ook naar de bedrijfsnummers (*ID* in de tabel *Corporation*, dat op zijn beurt correspondeert met *CORPORATION* in de tabel *Corporation_name*) opgenomen in de databank. Dit zijn dezelfde bedrijfsnummers als de nummers waaraan de aandelen of obligaties gekoppeld zijn (zie verder). Natuurlijk worden ook de bijbehorende data (*STARTDATE* - *ENDDATE*), dus de periode waarbinnen deze gegevens van kracht zijn, opgeslagen.



Dit alles vormt meteen **het unieke** aan het SCOB-systeem. **De bedrijven vormen de kern**. Als iemand dus een bedrijf opvraagt, krijgt hij onmiddellijk **alle beschikbare info**. Meer uitleg hierover is terug te vinden in **Deel II, Modules**, onder "De Bedrijfsmodule".



De databank is niet alleen uitbreidbaar met tabellen die gelinkt zijn aan de bedrijven. Er bestaat ook de mogelijkheid om allerlei soorten van algemene economische informatie op te

slaan in aparte tabellen, zoals bijvoorbeeld over inflatie, het bruto binnenlands product, ...

Op dit moment zitten er reeds een aantal verschillende archieven in de databank. Deze hebben allemaal benamingen in de stijl van "A. 1", "B. 1", "C01" enz. Elke letter verwijst naar de eerste letter van het uitgevende bedrijf, elk nummer naar een chronologische opeenvolging. Op deze manier zijn de documenten eenvoudig terug te vinden in de archiefdozen en mappen.

Er bestaat op dit ogenblik een volledige inventaris (die dezelfde manier van noteren volgt) in Word, die binnenkort gepubliceerd zal worden. De bedoeling is dat deze ook in de databank wordt ingevoerd en geraadpleegd kan worden d.m.v. "De Documentmodule" (**Deel II, Modules**).

5.2. Archiver (Documentmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME		VARCHAR2(200)
SESSIONPART		NUMBER

De tabel *Archiver* beschrijft de personen/instellingen die de archieven uit de vorige tabel *Archivenummer* "bijgehouden" hebben. Voor SCOB zijn dit bijvoorbeeld "de Beurs van Brussel", "de Beurs van Antwerpen", "de Bankcommissie", "SCOB" zelf, ...

Wat er hier ID genoemd wordt, correspondeert met ARCHIVER uit de tabel *Archivenummer*. De naam die correspondeert met dit uniek nummer wordt opgeslagen in NAME. Het attribuut SESSIONPART zou moeten dienen om een eerste "doos" met gegevens (ID uit de tabel *Sessionpart*, zie verder) aan de desbetreffende "archiver" te koppelen, en dit om een boomstructuur te maken die tot uiting komt in "de Documentmodule" (zie **Deel II, Modules**). Om deze boomstructuur in de praktijk te maken, steunt men echter **niet** meer op *Sessionpart*, maar gaat er hoofdzakelijk gebruik gemaakt worden van de tabellen *Archivenummer* (5.1.) en *Subarchivenummer* (5.61.). De tabel *Archivenummer* staat als het ware voor de "dozen" waarin alle archieven worden bijgehouden. *Subarchivenummer* staat voor de "mappen" en eventuele "submappen" in die dozen. D.m.v. de tabel *Document_san* worden aan elk "subarchieff"

(map of submap) de corresponderende documenten gekoppeld. In de tabellen `Document` en `Document_title` is dan bijkomende informatie over elk document terug te vinden. Voor nog meer informatie verwijzen we naar de desbetreffende tabellen. Een voorbeeld is in detail uitgewerkt onder (5.61.).

Op dit moment zitten er in de tabel `Archiver` een aantal gegevens die natuurlijk nog uitgebreid kunnen (en ook gaan) worden naar de andere instellingen die hun collectie archieven ter beschikking van SCOB stellen.

Een kleine illustratie van wat er nu reeds in de databank aanwezig is:

Archiver		
ID	NAME	SESSIONPART
---	-----	-----
1	Beurs van Brussel	1
2	SCOB prospectuscollectie	6

De nummers 1 en 6 (`SESSIONPART`) corresponderen dus met `ID` uit de tabel `Sessionpart`. Daar staan deze voor "dossiernummer". Zij worden echter niet meer gebruikt omdat er ondertussen een krachtigere en betere manier gevonden is om hun functie over te nemen. Voor meer uitleg hierover, zie de tabel `Sessionpart` (5.44.).

5.3. Bank (Bedrijfsmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(250)

De tabel `Bank` beschrijft de banken/financiële instellingen die de financiële diensten voor de bedrijven verzorgen.

`ID` geeft een uniek nummer aan elke bank, waarvan in `NAME` de volledige naam wordt opgeslagen.

We geven een sfeerbeeld (een willekeurige selectie van een aantal tupels uit de tabel `Bank`, om een idee te krijgen van wat er in de databank zit).

Bank	
ID	NAME
-----	-----
101	Siège Social de l'entreprise
100	Société Générale pour Favoriser l'industrie générale
102	Banque Auxiliaire
103	Banque de la Sambre
104	Banque Belge pour l'industrie à Bruxelles
105	Banque Dubois à Liège
106	Banque de Commerce à Anvers
107	Banque de la Société Générale de Belgique
108	Banque de Bruxelles
109	Banque Nagelmaeckers
110	Banque O.De Schaetzen et Cie
111	Société Française de Banque et de Dépôts

5.4. Bookkeepingitem (Bedrijfsmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(500)

In tabel `Bookkeepingitem` worden de bookkeepingitems bijgehouden. Deze tabel handelt over de Activa-Passiva posten en zit nog in een experimenteel stadium. De tabel zal hoofdzakelijk in "de Bedrijfsmodule" gebruikt worden (zie **Deel II, Modules**).

Bookkeepingitem	
ID	NAME
-----	-----
100	Capital
101	Fonds de Réserve Légale
102	Créditeurs divers
103	Portefeuille
104	Débiteurs Divers
105	Mobilier

5.5. Category (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(100)

In tabel `Category` worden de verschillende sectoren waarin wij de bedrijven plaatsen, opgeslagen onder een unieke nummer (`ID`), waaraan de naam (`NAME`) gekoppeld is. Om er enkele te noemen: "scheepvaart", "spoorwegen", "visserij" en "steenkool".



Let op, de categorieën die er hier gemaakt zijn door SCOB zullen bijna altijd **afwijken** van de sectoren die de beurs geeft. De SCOB-classificatie steunt op de NACE classificatie (een internationaal erkende classificatiemethode die overal ter wereld gebruikt wordt). De benaming "CATEGORY" en "SECTOR" dekken dus twee verschillende ladingen. "CATEGORY" is de SCOB-NACE sectorcode en "SECTOR" is de benaming van de entiteit waarbinnen de beurs een aandeel of obligatie lokaliseert.

5.6. Codetype (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME		VARCHAR2(200)

In tabel `Codetype` vinden we de verschillende types van codes terug onder een uniek nummer (`ID`), waaraan hun naam (`NAME`) gekoppeld is.

Dit zijn de verschillende codes die er bij de koerslijstnotaties in de loop der tijd gebruikt zijn, zoals de "ISIN-code" (International Securities Identification Numbering System), de "SVM/SRW-code" (Stock Valeurs Mobilières / Stock Roerende Waarden) en een interne code (`stockcode`), opgesteld voor het intern gebruik in de databank.

Vanaf 1969 vindt men in de koerslijsten voor elke stock de SVM-code (= SRW -code, de ene benaming is de Nederlandse versie, de andere de Franse) terug, en iets later verschijnt er ook de ISIN-code. Deze zijn per STOCK in de databank terug te vinden in de tabel `Stock_codetype` die aan elke stock zijn codetype en de daarbij horende code koppelt (voor meer info, zie verder). Deze codes dienen hoofdzakelijk om de elektronische transfer van koersen mogelijk te maken omdat de codes gebruikt worden als unieke identificatiemethode. Deze codetypes zijn wel temporeel veranderlijk, vandaar ook telkens het vermelden van `STARTDATE` en `ENDDATE`.

Stock_codetype				
STOCK	CODETYPE	CODE	STARTDATE	ENDDATE
----	-----	-----	-----	-----
25	2	3210	02-JAN-1969	02-JAN-1974
25	2	3210.09	02-JAN-1974	01-AUG-1989
36	2	3262	02-JAN-1969	02-JAN-1974
36	2	3262.61	02-JAN-1974	31-DEC-3999
74	2	3055	02-JAN-1969	02-JAN-1974
74	2	3489.94	02-JAN-1974	03-JUL-1989
74	2	3619.30	03-JUL-1989	31-DEC-3999
12829	1	BE0003798155	17-APR-2001	31-DEC-3999
12829	2	3798.15	17-APR-2001	31-DEC-3999
12829	3	SPA	17-APR-2001	31-DEC-3999

De verklaring van de codes is terug te vinden in `Codetype`.

Codetype	
ID	NAME
--	-----
1	ISIN Code
2	SRW (SVM) Code
3	Stock Code
4	Audiotex Code
5	Onbekend

5.7. Corporation (Bedrijfsmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

In tabel `Corporation` worden de verschillende bedrijven opgeslagen waarvan we info hebben. De bedrijven worden geïdentificeerd door een unieke nummer (`ID`), en we houden de startdatum (`STARTDATE`) bij wanneer het bedrijf begonnen is en de einddatum (`ENDDATE`).

Voor nog meer info verwijzen we naar **Deel II, Modules**, meerbepaald naar "de Bedrijfsmodule".

5.8. Corporationaction (Bedrijfsmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(250)
MULTIPLICITY	NOT NULL	NUMBER

In de tabel `Corporationaction` worden de verschillende fasen beschreven die een bedrijf kan doorlopen, zoals "oprichting", "overname", "splitsing", "naamsverandering",...

Deze tabel is niet meer operationeel. In feite is het een poging geweest om het "Logboek Aandelen" in de databank in te voeren, maar dit is te ingewikkeld gebleken. Het logboek is nu geschreven in Word (zie **Deel II, Modules**). Toch worden deze tabellen nog niet uit de databank verwijderd. Er wordt namelijk nog gewerkt aan een betere versie om dit toch te kunnen realiseren.

5.9. Corporationset (Bedrijfsmodule)

Column Name	Null?	Type
-----	-----	----
ID	NOT NULL	NUMBER
CORPORATION	NOT NULL	NUMBER
ROLE		NUMBER

De tabel Corporationset wordt niet meer gebruikt. Ook deze tabel maakte net zoals Corporationaction deel uit van een eerste "poging" om het "Logboek Aandelen" in de databank in te voeren. Ook deze tabel wordt nog niet verwijderd omdat er nog gewerkt wordt aan een betere versie.

5.10. Corporation_adminlocation (Bedrijfsmodule)

Column Name	Null?	Type
-----	-----	----
CORPORATION	NOT NULL	NUMBER
ADDRESS		VARCHAR2(250)
CITY		VARCHAR2(250)
COUNTRY		VARCHAR2(250)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel Corporation_adminlocation beschrijft de locatie van de **hoofdzetel** van het bedrijf.

Er wordt per bedrijfsnummer (CORPORATION) het adres (ADDRESS, CITY, COUNTRY) bijgehouden en de data van waarop het bedrijf zich daar vestigde (STARTDATE) en voor hoe lang (ENDDATE). Om te weten over welk bedrijf het hier gaat gebruiken we de tabel Corporation_name (zie verder). CORPORATION in Corporation_adminlocation correspondeert met CORPORATION in Corporation_name, waaruit dan de volledige naam gehaald kan worden (NAME).

5.11. Corporation_bank (Bedrijfsmodule)

Column Name	Null?	Type
CORPORATION	NOT NULL	NUMBER
BANK	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Corporation_bank` beschrijft de bank waarmee het bedrijf (CORPORATION) nauw samenwerkt. Deze wordt voorgesteld d.m.v. het uniek identificatienummer BANK.

BANK uit `Corporation_bank` correspondeert met ID uit `Bank`. Op deze manier wordt er aan elk nummer de bijbehorende naam van de bank (NAME in de tabel `Bank`) gelinkt.

5.12. Corporation_bookkeeping (Bedrijfsmodule)

Column Name	Null?	Type
CORPORATION	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ACTIVEPASSIVE	NOT NULL	CHAR(1)
AMOUNT	NOT NULL	NUMBER

In tabel `Corporation_bookkeeping` wordt er per bedrijf (CORPORATION) bijgehouden "hoe het nu precies zit met de boekhouding". Er wordt opgeslagen of het gaat over activa of passiva en wat het precieze bedrag is. Ook deze tabel wordt niet meer gebruikt. Er bestaat een verbeterde versie die de functies van deze tabel volledig overneemt, `Corporation_bookkeepingitem`.

5.13. Corporation_bookkeepingitem (Bedrijfsmodule)

Column Name	Null?	Type
CORPORATION	NOT NULL	NUMBER
ACTIVEPASSIVE	NOT NULL	CHAR(1)
STARTDATE	NOT NULL	DATE
COMMENTS	NOT NULL	NUMBER
AMOUNT	NOT NULL	NUMBER

De tabel Corporation_bookkeepingitem bevat essentieel dezelfde info als zijn oudere en niet meer gebruikte broertje Corporation_bookkeeping (zie 5.12.). Er is nu extra ruimte voorzien voor opmerkingen (COMMENTS).

Deze tabel zit nog in de **testfase**. Ze is nog nooit gebruikt geworden in de praktijk, omdat SCOB zich eerst op andere dingen (koerseninvoer, ...) heeft geconcentreerd. Omwille van deze reden is de vorige poging Corporation_bookkeeping dan ook nog niet verwijderd.

5.14. Corporation_capitalchanges (Bedrijfsmodule)

Column Name	Null?	Type
ID	NOT NULL	NUMBER
CORPORATION	NOT NULL	NUMBER
COMMENTS		VARCHAR2(2000)
INFODATE	NOT NULL	DATE
AMOUNT		NUMBER
EXACT	NOT NULL	CHAR(3)

In tabel Corporation_capitalchanges worden de kapitaalsveranderingen van de bedrijven opgeslagen. Elke verandering krijgt een eigen nummer (ID). Het bedrijf (CORPORATION) wordt bijgehouden, net als eventuele opmerkingen (COMMENTS), de datum van deze nieuwe informatie (INFODATE), het bedrag waarover het gaat (AMOUNT) en of dit exact is (EXACT).

5.15. Corporation_evolution (Bedrijfsmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
OLD		NUMBER
NEW		NUMBER
ACTION	NOT NULL	NUMBER
NOTARY		VARCHAR2(250)
NOTARYDATE		VARCHAR2(250)
GOVDATE		VARCHAR2(250)
GOVDOCUMENT		VARCHAR2(250)
COMMENTS		VARCHAR2(2000)

De tabel `Corporation_evolution` beschrijft per bedrijfsnummer (ID) de evolutie van het bedrijf. Dit is vooral van belang als een bedrijf opgeslokt wordt door een ander bedrijf of van naam verandert. De precieze oorzaak is terug te vinden onder `ACTION`. Dit attribuut beschrijft dus wat er precies met het bedrijf gebeurde. Verder wordt er bij elke gebeurtenis die er plaatsvindt in de loop van de geschiedenis van een bedrijf extra informatie bijgehouden over de juridische kant van de zaak (`NOTARY`, `NOTARYDATE`). Ook de documentatie en dat datum dat alles publiek gemaakt werd, wordt opgeslagen (`GOVDATE`, `GOVDOCUMENT`). Verder is het ook mogelijk om extra uitleg of commentaar bij de gebeurtenis in te voeren (`COMMENTS`).

5.16. Corporation_information (Bedrijfsmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
CORPORATION	NOT NULL	NUMBER
COMMENTS		VARCHAR2(4000)
INFODATE	NOT NULL	DATE

De tabel `Corporation_information` bevat per bedrijfsnummer (`CORPORATION`) opmerkingen (`COMMENTS`). Ook de datum waarop men over deze info beschikt wordt opgeslagen (`INFODATE`).

5.17. Corporation_location (Bedrijfsmodule)

Column Name	Null?	Type
CORPORATION	NOT NULL	NUMBER
ADDRESS		VARCHAR2(200)
CITY	NOT NULL	VARCHAR2(200)
COUNTRY		VARCHAR2(200)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Corporation_location` geeft voor elk bedrijf (`CORPORATION`) de vestigingsplaats van zijn **maatschappelijke** of **juridische** zetel (`ADDRESS`, `CITY`, `COUNTRY`) en de data vanaf wanneer dat dit zo is (`STARTDATE`) en voor hoelang (`ENDDATE`).

Er is dus wel degelijk een verschil tussen de tabellen `Corporation_location` (5.17.) en `Corporation_adminlocation` (5.10.)

5.18. Corporation_name (Bedrijfsmodule)

Column Name	Null?	Type
CORPORATION	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(200)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Corporation_name` geeft voor elke uniek bedrijfsnummer (`CORPORATION`) de naam (`NAME`) van het bedrijf dat hiermee geïdentificeerd wordt in de databank en de tijdsspanne wanneer deze benaming gebruikelijk was (`STARTDATE` - `ENDDATE`).

Er zitten op dit moment meer bedrijfsnamen in de databank dan er bedrijven in de databank zitten (in de tabel `Corporation`). Dit wijst erop dat er in de loop van tijd een aantal bedrijven van naam veranderd zijn, bijvoorbeeld ten gevolge van een fusie (zie `Corporationaction` voor een volledig overzicht van wat er allemaal met een bedrijf kan gebeuren waardoor het van hoedanigheid verandert).

5.19. Corporation_resultitems (Bedrijfsmodule)

Column Name	Null?	Type
CORPORATION	NOT NULL	NUMBER
PROFITLOSS	NOT NULL	CHAR(1)
STARTDATE	NOT NULL	DATE
COMMENTS	NOT NULL	NUMBER
AMOUNT	NOT NULL	NUMBER

De tabel `Corporation_resultitems` beschrijft per bedrijf (`CORPORATION`) of er winst gemaakt is of verlies (`PROFITLOSS`). Dit wordt aangegeven met de letters Y of N. Tevens wordt de startdatum (`STARTDATE`) van dit gegeven bijgehouden, net als eventuele bemerkingen (`COMMENTS`) en het desbetreffende bedrag (`AMOUNT`).

5.20. Corporation_results (Bedrijfsmodule)

Column Name	Null?	Type
CORPORATION	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
PROFITLOSS	NOT NULL	CHAR(1)
AMOUNT	NOT NULL	NUMBER

De tabel `Corporation_results` geeft per bedrijf (`CORPORATION`) de credit (kosten)/ debet (opbrengsten) van de resultatenrekening (`PROFITLOSS`). Tevens worden de startdatum (`STARTDATE`) en het bedrag (`AMOUNT`) van deze gebeurtenis bijgehouden.

Deze tabel is eigenlijk de voorloper van de tabel `Corporation_resultitems`. Deze tabel bevat essentieel dezelfde info, maar kan ook nog extra opmerkingen (`COMMENTS`) verwerken. Daarom is de tabel `Corporation_resultitems` een verbetering van `Corporation_results`, die dan ook niet meer gebruikt zal worden in de toekomst.

5.21. Corporation_sociallocation (Bedrijfsmodule)

Column Name	Null?	Type
CORPORATION	NOT NULL	NUMBER
ADDRESS		VARCHAR2(250)
CITY		VARCHAR2(250)
COUNTRY		VARCHAR2(250)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Corporation_sociallocation` geeft voor elk bedrijf (`CORPORATION`) de locatie (`ADDRESS`, `CITY`, `COUNTRY`) van zijn **administratieve zetel** en de corresponderende periode (`STARTDATE` - `ENDDATE`) dat dit van kracht was.

5.22. Corporation_stockholders (Bedrijfsmodule)

Column Name	Null?	Type
ID	NOT NULL	NUMBER
STOCK	NOT NULL	NUMBER
CORPORATION		NUMBER
PERSON		NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE
COMMENTS		VARCHAR2(500)
PERCENTAGE		NUMBER
AMOUNT		NUMBER
NUMOFSTOCKS		NUMBER

De tabel `Corporation_stockholders` geeft aan per aandeel (`STOCK`) van een bedrijf (`CORPORATION`) wie (`PERSON`) dat aandeel "bezit". De geldigheidsdata (`STARTDATE` - `ENDDATE`) worden bijgehouden, evenals eventuele opmerkingen (`COMMENTS`), het exacte bedrag (`AMOUNT`) of percentage (`PERCENTAGE`) en het aantal aandelen (`NUMOFSTOCKS`). Voor elk van deze reeks gegevens wordt een uniek nummer bijgehouden (`ID`).

5.23. Corporation_stocks (Bedrijfsmodule)

Column Name	Null?	Type
CORPORATION	NOT NULL	NUMBER
STOCK	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE
COMMENTS		VARCHAR2(500)
PERCENTAGE		NUMBER
AMOUNT		NUMBER
NUMOFSTOCKS		NUMBER

De tabel `Corporation_stocks` geeft per bedrijf (`CORPORATION`) welke aandelen (`STOCK`) er uitgegeven zijn. Verder worden er dezelfde gegevens bijgehouden als in `Corporation_stockholders`, zoals de tijdsperiode dat dit van kracht was (`STARTDATE` - `ENDDATE`), eventuele opmerkingen (`COMMENTS`), of het over een percentage gaat (`PERCENTAGE`), het juiste bedrag (`AMOUNT`) en het aantal stocks (`NUMOFSTOCKS`).

5.24. Currency (Aandeelmodule)

Column Name	Null?	Type
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(100)

In de tabel `Currency` wordt er aan elke munteenheid (`NAME`) een uniek nummer (`ID`) gekoppeld.

Een klein voorbeeldje:

Currency	
ID	NAME
1	Frank (BELGIQUE)
2	Gulden (fl) (Nederland) (fl = florijn) (PAYS BAS)
3	Lire (of lira) (Italiaanse) (ITALIE)
4	Kredietroebel of zilverroebel (Roebel) (RUSSIE)(PLUSIEURS) (Rouble d'argent)
5	Dollar (USA) (\$)

5.25. Dividendtype (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(100)

De tabel `Dividendtype` geeft aan elke soort dividend (`NAME`) een uniek identificatienummer (`ID`).

Een klein voorbeeldje:

Dividendtype	
ID	NAME
----	-----
1	Eerste dividend (Première Dividende) (1° div)
2	Tweede dividend (Deuxième Dividende) (2° div)
3	Dividend (en/of) Interest (Dividende/Intérêt)
4	DIVIDENDE NETTO !
5	Halve interest (1/2 Intérêt)
6	Halve interest plus dividend (1/2 intérêt + Dividende)
7	1ste + 2de semesterdividend (samen uitbetaald) (1° + 2° div.: même date)
8	Répartition (en liquidation)

5.26. Document (Documentmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
AUTHOR		VARCHAR2(200)
PAGES		NUMBER
PUBLISHDATE		DATE
LOCATION		VARCHAR2(200)
PUBLISHER		NUMBER
TYPE		NUMBER
ABSTRACT		VARCHAR2(2000)
EXACTDATE		CHAR(1)

De tabel `Document` geeft aan elk document dat het SCOB ter beschikking heeft een uniek nummer (`ID`), en houdt de auteur (`AUTHOR`) bij, het aantal bladzijden waaruit het document bestaat (`PAGES`), de datum van publicatie (`PUBLISHDATE`), de uitgever (`PUBLISHER`,

correspondeert met ID in de tabel Corporation, dus staat voor een bedrijf), het soort document (TYPE, correspondeert met ID uit de tabel Documenttype), een korte beschrijving van de inhoud (ABSTRACT) en een karakter (Y/N) om aan te duiden of de datum exact is of niet (EXACTDATE).

Om te weten welk document elk nummer vertegenwoordigt, heeft men de tabel Document_title (zie verder) nodig. ID uit Document correspondeert met DOCUMENT uit Document_title, waaruit dan de volledige naam (NAME) afgelezen kan worden.

5.27. Documenttype (Documentmodule)

Column Name	Null?	Type
-----	-----	----
ID	NOT NULL	NUMBER
NAME		VARCHAR2(200)

De tabel Documenttype geeft aan elk type van document (NAME) dat er kan voorkomen een uniek identificatienummer (ID).

Een illustratie:

Documenttype	
ID	NAME
----	-----
1	statuut
2	balans/resultaten
3	jaarverslag
4	correspondentie
5	prospectus
6	wettekst
7	reglementen
8	brief

5.28. Document_san (Documentmodule)

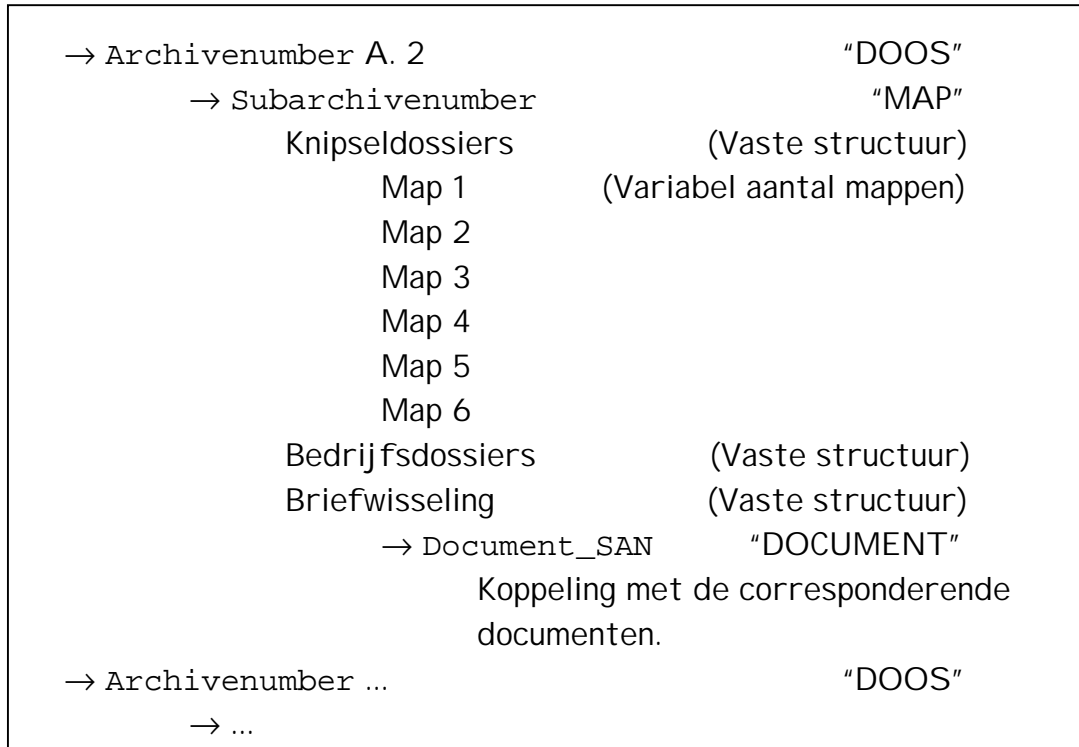
Column Name	Null?	Type
DOCUMENT	NOT NULL	NUMBER
SAN	NOT NULL	NUMBER

De tabel `Document_san` maakt een wederzijdse koppeling tussen documenten (`DOCUMENT`) en "SAN". SAN staat voor **SubArchiveNumber**. Deze tabel koppelt dus aan elke "subarchief" de daarin zittende "documenten". Je kan een subarchief dus best zien als een "map" waarin verschillende "documenten" zitten.

Op deze manier kan er een soort van boomstructuur gemaakt worden die de hiërarchie tussen de archieven weerspiegelt.

We kunnen dit schetsen met een klein voorbeeldje:

Beurs van Brussel	
→ Archivenumber A. 1	"DOOS"
→ Subarchivenumber	"MAP"
Knipseldossiers	(Vaste structuur)
Map 1	(Variabel aantal mappen)
Map 2	
Bedrijfsdossiers	(Vaste structuur)
Briefwisseling	(Vaste structuur)
→ Document_SAN	"DOCUMENT"
Aan "Map1", "Map2", "Bedrijfsdossiers" en "Briefwisseling" worden de corresponderende documenten gekoppeld. De benaming en verdere eigenschappen kunnen gevonden worden in de tabellen <code>Document</code> en <code>Document_title</code> .	



5.29. Document_title (Documentmodule)

Column Name	Null?	Type
DOCUMENT	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(220)

De tabel `Document_title` geeft bij elk identificatienummer van een document (`DOCUMENT`) zijn officiële naam (`NAME`).

Het aantal documenten die er ter beschikking van de gebruikers van de databank staan correspondeert met het aantal gekoppelde (dus in "mappen" geplaatste) documenten in `Document_SAN` (5.28.).

5.30. Evolutionrole (Bedrijfsmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(250)

De tabel `Evolutionrole` beschrijft de verschillende soorten evoluties die een bedrijf kan ondergaan. Elke evolutie krijgt een uniek nummer (`ID`) dat aan zijn specifieke omschrijving (`NAME`) wordt gekoppeld.

5.31. Job (Bedrijfsmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(250)

De tabel `Job` koppelt aan elk "beroep", elke "hoedanigheid" of "functie" die een persoon kan vervullen in een bedrijf (`NAME`) een uniek identificatienummer (`ID`).

Een aantal mogelijke beroepen die er terug te vinden zijn:

Job	
ID	NAME
-----	-----
100	administrateur
101	administrateur-délégué
102	Président
103	Commissaire
104	Secrétaire
105	Aandehouder

5.32. Notation (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
STOCK	NOT NULL	NUMBER
SECTOR	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Notation` is voor "de Aandeelmodule" een van de **belangrijkste** tabellen die er in de databank aanwezig is.

In deze tabel wordt per aandeel (`STOCK`) bijgehouden vanaf wanneer en voor hoelang (`STARTDATE` - `ENDDATE`) dat aandeel in een zekere sector (`SECTOR`) genoteerd staat. Elke groep van zulke gegevens krijgt een uniek identificatienummer (`ID`), de "notatie".

Deze tabel vol "notaties" is een directe weerspiegeling van de gegevens die er in de koerslijsten terug te vinden zijn.

5.33. Notation_price (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
NOTATION	NOT NULL	NUMBER
DAY	NOT NULL	DATE
ARGENT		NUMBER
PAPIER		NUMBER
OPEN		NUMBER
CLOSE		NUMBER
MIN		NUMBER
MAX		NUMBER
PREVIOUS		NUMBER
EXTERN		NUMBER
PERCENT		CHAR (1)

De tabel `Notation_price` mag met recht en rede de **belangrijkste** tabel voor "de Aandeelmodule" genoemd worden. Deze tabel bevat spectaculair veel gegevens en groeit per dag gestadig aan. Dat komt omdat in deze

tabel alle koersen uit de koerslijsten ingevoerd worden. Het is bijgevolg "het kloppend hart" van "de Aandeelmodule" (zie **Deel II, Modules**).

In deze tabel worden per dag (DAY) en per notatie (NOTATION uit Notation_price correspondeert met ID uit Notation) de **koersen** van een zeker aandeel ingevoerd. De gegevens komen rechtstreeks uit de koersboeken door de Beurs van Brussel gepubliceerd (en naarmate de datum in de jaren '90 terechtkomt uit de Financieel Economische Tijd) en zijn manueel door jobstudenten in de databank ingevoerd. Tot nu is er door de enorme hoeveelheid aan koersgegevens enkel gewerkt met het invoeren van de koersen op de laatste dag van de maand.

De koersen van aandelen staan bijna uitsluitend in absolute waarden, de koersen van obligaties komen terug in absolute waarden, maar ook in procenten. Wat betreft de invoer in de databank hebben we er voor gezorgd dat de koersen van aandelen volledig zijn opgenomen als reële prijzen in BEF, van 1832 t.e.m. 1999. Vanaf dat moment zijn ze uitgedrukt in EURO.

De koersen zelf kunnen van verschillende aard zijn. We hebben in de databank de mogelijkheden ARGENT en PAPIER (bied- en laatkoersen), OPEN, CLOSE (slot), MIN, MAX, PREVIOUS (de vorige koers) en EXTERN (als de koers niet uit het koersboek komt, maar uit een externe bron) voorzien. PERCENT geeft aan of het in reële waarden dan wel in procent is uitgedrukt.

De koersen werden dus handmatig ingevoerd door jobstudenten. In "de Aandeelmodule" werden speciale procedures ingebouwd die de foutieve invoer van koersen moeten minimaliseren. Zo is er een automatische detectie ingebouwd die een waarschuwing geeft als een ingevoerde koers meer dan 10% afwijkt van de vorige koers die er reeds in de databank zit. Er is dus, zoals reeds eerder opgemerkt, slechts gewerkt met het opnemen van koersen op maandbasis. We hebben steeds de koers genomen op de laatst beschikbare dag van de maand. Bij voldoende financiële ondersteuning kunnen in een later stadium koersen op dagbasis worden ingevoerd.

5.34. Notation_tradedquantity (Aandeelmodule)

Column Name	Null?	Type
NOTATION	NOT NULL	NUMBER
DAY	NOT NULL	DATE
VOLUME	NOT NULL	NUMBER

De tabel `Notation_tradedquantity` is een tabel die nauw verbonden is met de twee vorige tabellen `Notation` en `Notation_price`.

In deze tabel wordt er per notatie (`NOTATION`) en per dag (`DAY`) de hoeveelheid (`VOLUME`) van de verhandelde financiële waarde genoteerd.

5.35. Person (Bedrijfsmodule)

Column Name	Null?	Type
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(250)

De tabel `person` geeft aan elke persoon die er in de databank terecht komt (`NAME`) een uniek identificatienummer (`ID`).

Een klein voorbeeldje:

Person	
ID	NAME
---	-----
100	Lippens H.
101	Thys Albert
102	Urban Jules
103	De Laveleye Georges
104	Despret Ed.
105	Brugmann Georges

5.36. Person_function (Bedrijfsmodule)

Column Name	Null?	Type
PERSON	NOT NULL	NUMBER
JOB	NOT NULL	NUMBER
CORPORATION	NOT NULL	NUMBER
COMMENTS		VARCHAR2(4000)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Person_function` beschrijft voor elke `persoonsID` (`PERSON`, correspondeert met `ID` uit `Person`, waaruit de volledige naam terug te vinden is) zijn jobfunctie (`JOB`, correspondeert met `ID` uit de tabel `Job`, waarin ook de volledige omschrijving van het beroep is terug te vinden) bij het desbetreffende bedrijf (`CORPORATION`), de corresponderende data (`STARTDATE` - `ENDDATE`) en eventuele opmerkingen (`COMMENTS`).

5.37. Person_location (Bedrijfsmodule)

Column Name	Null?	Type
PERSON	NOT NULL	NUMBER
ADDRESS		VARCHAR2(250)
CITY		VARCHAR2(250)
COUNTRY		VARCHAR2(250)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Person_location` geeft per unieke `persoonsid` (`PERSON`, correspondeert met `ID` uit de tabel `Person` waarin de volledige naam van de persoon is terug te vinden) de verblijfplaats (`ADDRESS`, `CITY`, `COUNTRY`) van die persoon en de tijdsspanne waarbinnen deze geldig is (`STARTDATE` - `ENDDATE`).

5.38. Quantitytype (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(100)

De tabel `Quantitytype` geeft aan elke type hoeveelheid (`NAME`) een unieke identificatiesleutel (`ID`). Het gaat hier over types in de stijl van "tot de beurs toegelaten titels", "bestaande effecten", ...

Een voorbeeld:

ID	NAME
-----	-----
1	Uitgegeven titels ("titres émis")
2	Tot de beurs toegelaten titels ("titres admis")
3	Titels in circulatie ("titres en circulation")
4	Niet-volstorte waarde ("à verser")
5	Gepland aantal aandelen
6	Bestaande effecten ("titres existants")

5.39. Resultitems

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(500)

De tabel `Resultitems` geeft aan elk item (`NAME`) een unieke identificatiesleutel (`ID`).

5.40. Sector (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

Eenmaal een beurs gecreëerd is, worden er sectoren gemaakt die aan deze beurs gekoppeld zijn en die alzo terug te vinden zijn in de koerslijsten. Deze krijgen in `Sector` een uniek identificatienummer (ID). Het begrip "sector" moet breed opgevat worden en is eigenlijk te vergelijken met de hoofdstukken en paragrafen in de indeling van elk boek. Voor het duidelijke verschil in opvattingen tussen "sector" en "category" verwijzen we naar 5.5.

De tabel `Sector` is nog een van de belangrijke tabellen voor het SCOB. In deze tabel worden naast de ID's (ID) de levensduren van de sectoren opgeslagen (STARTDATE - ENDDATE). Eenzelfde sector (geïdentificeerd door dezelfde ID, waarvan de bijbehorende naam terug te vinden is in de tabel `Sector_name`, waarin `SECTOR` correspondeert met ID uit `Sector`) kan gedurende verschillende tijdsspannes hebben bestaan. Indien de einddatum niet gekend zou zijn, wordt er standaard 31/12/3999 gebruikt. Een voorbeeld:

Sector		
ID	STARTDATE	ENDDATE
---	-----	-----
66	03-JAN-1832	31-DEC-3999
67	03-JAN-1832	03-JAN-1881
68	03-JAN-1832	31-DEC-3999
69	03-JAN-1832	03-JAN-1966

Om de verschillende beurzen van elkaar te onderscheiden, gebruiken we voor de eerste beurs (ID = 1) nummers vanaf 1 (er staan hier vier voorbeelden van, sectoren met ID 66, 67, 68 en 69), voor de tweede beurs (ID = 2) vanaf 1000, voor de derde beurs (ID = 3) vanaf 10 000 enz.

Ter vervollediging geven we hier nu nog de benamingen (en hun veranderingen in de tijd) van de sectornummers, dus de omschrijving van

de sectoren die in de databank terug te vinden zijn onder de nummers 66, 67, 68 en 69:

Sector_name			
SECTOR	NAME	STARTDATE	ENDDATE
-----	-----	-----	-----
66	Fonds Publics (Brussel)	03-JAN-1832	15-JUL-1861
66	Valeurs Etrangères (Brussel)	15-JUL-1861	15-FEB-1869
66	Fonds d'Etats et de Villes Etrangères (Brussel)	15-FEB-1869	07-JAN-1880
66	Fonds d'Etats et de Villes (Brussel)	07-JAN-1880	31-DEC-3999
67	Actions et Obligations (Brussel)	03-JAN-1832	02-JAN-1861
67	Actions (Brussel)	02-JAN-1861	03-JAN-1881
68	Banques, Caisses etc (Brussel: actions)	03-JAN-1832	26-OCT-1880
68	Banques, Caisses et Sociétés Immobilières (Brussel: actions)	26-OCT-1880	23-FEB-1898
68	Banques, Assurances et Entreprises Immobilières (Brussel: actions)	23-FEB-1898	17-AUG-1909
68	Assurances, Banques et Entreprises Immobilières (Brussel: actions)	17-AUG-1909	09-NOV-1925
68	Assurances, Banques (Brussel: actions)	09-NOV-1925	04-MAR-1935
68	Banques de Dépôts et de Crédit (Brussel: actions)	04-MAR-1935	31-DEC-3999
69	Chemins de fer (Brussel: actions)	03-JAN-1832	02-JAN-1861
69	Chemins de fer et Canaux (Brussel: actions)	02-JAN-1861	20-NOV-1933
69	Chemins de fer et Transports par Eau (Brussel: actions)	20-NOV-1933	31-DEC-3999

5.41. Sector_child (Aandeelmodule)

Column Name	Null?	Type
SECTOR	NOT NULL	NUMBER
CHILD	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

Deze tabel `sector_child` is erg verbonden met de tabel `sector`. Hij is speciaal opgesteld in de databank om het mogelijk te maken om een soort "boomstructuur" op te stellen van sectoren die **deelsectoren** (**subsectoren**) zijn van andere sectoren (**hoofdsectoren**). Deze boomstructuur wordt gevisualiseerd in "de Aandeelmodule" (zie **Deel II, Modules**). In de tabel `sector_child` wordt enkel de eerste subsector aan de hoofdsector gekoppeld. Voor de volgende sectoren op hetzelfde niveau wordt gebruik gemaakt van `sector_next`.

5.42. Sector_name (Aandeelmodule)

Column Name	Null?	Type
SECTOR	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(250)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

Deze tabel `sector_name` moet bekeken worden samen met de tabel `sector`. Bij elke `SECTOR` (correspondeert met ID uit `sector`) hoort de bijbehorende naam (`NAME`) en de periode waarin deze naam geldig was (`STARTDATE` - `ENDDATE`). De constructie met `STARTDATE` en `ENDDATE` zorgt er voor dat de naamsveranderingen van eenzelfde sector in de databank opgenomen kunnen worden. De unieke identificatiesleutel van een sector (`SECTOR`) blijft hierbij hetzelfde, maar zijn naam en de periode dat deze van kracht is wijzigen.

Er zijn meer sectornamen in de databank opgenomen dan er sectoren zijn, dus er zijn in de loop der tijd reeds een aantal naamsveranderingen van sectoren in de koerslijsten doorgevoerd.

We hernemen nog even een gedeelte van het vorige voorbeeld (onder 5.40.) waarbij we ons in het bijzonder toespitsen op sector 66.

Sector_name			
SECTOR	NAME	STARTDATE	ENDDATE
-----	-----	-----	-----
66	Fonds Publics	03-JAN-1832	15-JUL-1861
66	Valeurs Etrangères	15-JUL-1861	15-FEB-1869
66	Fonds d'Etats et de Villes Etrangères	15-FEB-1869	07-JAN-1880
66	Fonds d'Etats et de Villes	07-JAN-1880	31-DEC-1899

Hieruit blijkt duidelijk het **temporeel veranderlijke karakter van de data**. De sectorsleutel (SECTOR) is weliswaar uniek (in dit voorbeeld is 66 gekozen) maar achtereenvolgens kan men hier verschillende namen aan toewijzen (NAME). De periodes (STARTDATE - ENDDATE) waarin een bepaalde naam geldig is, hoeven niet noodzakelijk op elkaar aan te sluiten. Ze kunnen wel nooit overlappen. Twee verschillende namen binnen één periode is dus niet mogelijk. Een naam is immers uniek gebonden aan een periode. Een periode waarin er geen naam is, kan wel. Dat zou betekenen dat er in die periode ook geen sprake is van "die" sector.

5.43. Sector_next (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
SECTOR	NOT NULL	NUMBER
NEXT	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Sector_next` moet bekeken worden samen met de tabel `Sector_child`. `Sector_next` geeft voor elke sector zijn **opvolger** aan in de "boomstructuur", die gevisualiseerd wordt in het linkerpaneel van de "Aandeelmodule" (zie **Deel II, Modules**). Deze tabel wordt gebruikt voor alle volgende sectoren die op hetzelfde niveau komen. Men geeft er dus in aan welke sectoren in de koersboekstructuur op dezelfde "hoogte" staan. De namen (want hier (SECTOR) werkt men met de unieke

identificatienummers (Type is immers NUMBER)) kan men terugvinden in de tabel `Sector_name` onder `NAME`. Hiertoe correspondeert `SECTOR` uit `Sector_next` met `SECTOR` uit `Sector_name`.

We geven een kleine illustratie aan de hand van de gegevens die we hebben over de Beurs van Brussel. Voor deze beurs geldt er dat de eerste sector (66) vanaf de eerste dag (3/1/1832) gevolgd wordt door sector 67, maar vanaf 2/1/1861 door sector 74. De `ENDDATE` moet dus geïnterpreteerd worden als "tot" en mag en dus niet bijgerekend worden. Welke sectoren deze nummers 66, 67 en 74 vertegenwoordigen is terug te vinden in de tabel `Sector_name`.

Sector_next			
SECTOR	NEXT	STARTDATE	ENDDATE
-----	-----	-----	-----
66	67	03-JAN-1832	02-JAN-1861
66	74	02-JAN-1861	15-JUL-1861
66	555	15-JUL-1861	15-FEB-1869
66	79	15-FEB-1869	26-DEC-1873
66	80	26-DEC-1873	03-JAN-1947
66	104	03-JAN-1947	04-MAY-1959
66	204	04-MAY-1959	30-JUL-1962
66	78	30-JUL-1962	31-JUL-1967
66	224	31-JUL-1967	20-APR-1976
66	242	20-APR-1976	31-DEC-3999

5.44. Sessionpart (Documentmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME		VARCHAR2(250)
RANK		NUMBER
PARENT		NUMBER

De tabel `Sessionpart` had als bedoeling om aan elke "sessie" (`NAME`) een unieke identificatiesleutel (`ID`) te koppelen. Een "sessie" is een soort van "overkoepelend" document, waaronder een aantal subdocumenten teruggevonden kunnen worden. Op die manier zou er een soort boomstructuur ontstaan waarbij "mappen" terug te vinden zijn in "dozen" en "knipsels" in "mappen" enz. De tabel `Sessionpart` wordt echter **niet**

meer gebruikt voor dit doel. Deze functie wordt overgenomen door de tabel `Subarchivenumber`. Voor meer info verwijzen we dan ook naar deze tabel (5.61.).

5.45. Sharetype (Aandeelmodule)

Column Name	Null?	Type
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(100)

De tabel `Sharetype` geeft een unieke identificatiesleutel (ID) aan elk type van de financiële activa (NAME).

Een illustratie:

Sharetype	
ID	NAME
1	Aandeel
2	Obligatie
3	Interest
4	Droits (rechten)
5	Opties
6	Warrant

5.46. Stock (Aandeelmodule)

Column Name	Null?	Type
ID	NOT NULL	NUMBER
SHARETYPE	NOT NULL	NUMBER
STOCKEXCHANGE	NOT NULL	NUMBER

Ook de tabel `Stock` is ook een van de belangrijke tabellen in de databank.

Deze tabel beschrijft voor elke stock (ID, de echte benaming kan teruggevonden worden in `Stock_name`, waarbij ID uit `Stock`

correspondeert met STOCK uit Stock_name) van welk type (SHARETYPE) (deze sleutel correspondeert met ID uit Sharetype, waaruit de volledige naam gehaald kan worden) dat deze is en op welke beurs hij genoteerd staat (STOCKEXCHANGE). Deze constructie zorgt er voor dat er meerdere beurzen in de databank betrokken kunnen worden en dat eenzelfde aandeel dat op meerdere beurzen staat niet verward kan worden.

5.47. Stockexchange (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel Stockexchange geeft voor elke beurs (hier genoteerd d.m.v. een unieke sleutel ID, die overeenkomt met STOCKEXCHANGE in Stockexchange_name, en waarvan de naam voluit teruggevonden kan worden onder NAME in diezelfde tabel) zijn levensduur (STARTDATE - ENDDATE).

Deze tabel dient dus voor de **identificatie van de beurs**. Dit maakt het mogelijk om informatie uit andere beurzen in één en dezelfde databank te behandelen.

Een voorbeeld van een aantal beurzen die er in de databank zitten:

Stockexchange		
ID	STARTDATE	ENDDATE
----	-----	-----
1	01-JAN-1000	31-DEC-3999
2	01-JAN-1000	31-DEC-3999
3	01-JAN-1000	31-DEC-3999
4	01-JAN-1000	31-DEC-3999
999	01-JAN-1000	31-DEC-3999

Men merkt dat standaard de STARTDATE op 01-JAN-1000 en de ENDDATE op 31-DEC-3999 gezet worden. We hebben hier bewust voor

gekozen omdat dit zeer duidelijk periodes zijn waarbinnen het beursgebeuren valt. Natuurlijk is het aantal beurzen nog uitbreidbaar.

Voor alle duidelijkheid voegen we hier nog eens de volledige namen van de in bovenstaand voorbeeld genoemde beurzen toe.

Stockexchange_name			
STOCKEXCHANGE	NAME	STARTDATE	ENDDATE
1	Brussel	01-JAN-1000	31-DEC-3999
2	Antwerpen	01-JAN-1000	31-DEC-3999
3	Parijs	01-JAN-1000	31-DEC-3999
4	Amsterdam	01-JAN-1000	31-DEC-3999
999	Niet genoteerd	01-JAN-1000	31-DEC-3999

5.48. Stockexchange_child (Aandeelmodule)

Column Name	Null?	Type
STOCKEXCHANGE	NOT NULL	NUMBER
SECTOR	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `stockexchange_child` is ontworpen om de "boomstructuur" van sectoren aan een beurs te koppelen.

In deze tabel wordt de allereerste sector van de koersboekstructuur aan een bepaalde beurs gekoppeld. Dit is een eenmalige operatie, zolang de eerste sector van de koersboekstructuur dezelfde blijft. Bijgevolg is aan het ID-nummer van de beurs (`STOCKEXCHANGE` in `Stockexchange_child` = ID in `Stockexchange`) een ID-nummer van de eerste sector (`SECTOR` in `Stockexchange_child` = ID in `Sector`) gekoppeld, met begin- en einddatum (`STARTDATE` - `ENDDATE`).

Op dit moment wordt er hoofdzakelijk gewerkt op de Beurs van Brussel (ID = 1), maar er is reeds geëxperimenteerd met andere beurzen, waaronder de Beurs Van Antwerpen (ID = 2).

Met een voorbeeld tonen we nu hoe **het temporele karakter** van de data tot uiting komt in deze tabel.

Voor de Beurs van Brussel wordt aan de ID van de beurs (ID = STOCKEXCHANGE = 1) de sector met nummer 66 als eerste sector gekoppeld en dit tot 15/7/1861. Daarna is niet langer sector 66 de eerste sector, maar wel sector 77. Deze wordt vanaf dan (en dit voor hoelang dat dit duurt, dus tot 23/10/1889) aan 1 gekoppeld. Zo wordt er verder gewerkt tot nu. De einddatum van de laatste periode wordt standaard op 31/12/3999 gezet omdat we natuurlijk nog niet weten hoe lang dat dit gaat voortduren.

De tabel `stockexchange_child` bevat dus per beurs slechts één koppeling op één moment. Als de allereerste sector in de koersboekstructuur verandert, wordt er een nieuwe koppeling doorgevoerd. Dit komt echter relatief zelden voor. Voor Brussel is dit tot vandaag de dag slechts 5 keer gebeurd.

Stockexchange_child			
STOCKEXCHANGE	SECTOR	STARTDATE	ENDDATE
-----	-----	-----	-----
1	66	03-JAN-1832	15-JUL-1861
1	77	15-JUL-1861	23-OCT-1889
1	271	23-OCT-1889	01-JAN-1992
1	295	01-JAN-1992	15-APR-2002
1	297	15-APR-2002	31-DEC-3999

5.49 Stockexchange_name (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
STOCKEXCHANGE	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(250)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `stockexchange_name` geeft aan elk uniek nummer voor een beurs (`STOCKEXCHANGE`) zijn volledige naam (`NAME`) en de tijdsspanne (`STARTDATE` - `ENDDATE`) dat deze wederzijdse identificatie geldig is.

Er wordt op dit moment hoofdzakelijk gewerkt met de Beurs van Brussel, maar over het algemeen staan er nu reeds andere beurzen vermeld, waaronder Antwerpen, Parijs, en Amsterdam.

Stockexchange_name			
STOCKEXCH.	NAME	STARTDATE	ENDDATE
-----	-----	-----	-----
1	Brussel	01-JAN-1000	31-DEC-3999
2	Antwerpen	01-JAN-1000	31-DEC-3999
3	Parijs	01-JAN-1000	31-DEC-3999
4	Amsterdam	01-JAN-1000	31-DEC-3999
999	Niet genoteerd	01-JAN-1000	31-DEC-3999

5.50. Stocktype (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(250)

De tabel `stocktype` geeft aan elk type van aandeel (`NAME`) een unieke identificatienummer (`ID`).

Een illustratie van een aantal types die er in gebruik zijn:

Stocktype	
ID	NAME
---	-----
10	action de capital (kapitaalaandeel)
11	action ordinaire (gewoon aandeel)
12	action garanti (aandeel met winstgarantiel)
13	action privilégiiée (bevoorrecht aandeel)
14	action prioritaire (prioritair aandeel)
15	action de préférence (preferent aandeel)
16	part de réserve (reserve aandeel)
17	part sociale (Maatschappelijk aandeel)
18	Actions (aandelen) (zonder verdere aanduiding)
19	action primitive (primitief aandeel)
20	part de bénéfice (winstaandeel)

5.51. Stock_category (aandeelmodule)

Column Name	Null?	Type
STOCK	NOT NULL	NUMBER
CATEGORY	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

Deze tabel `stock_category` koppelt aan elke `stock` (`STOCK`) een bepaalde categorie (`CATEGORY`) en de bijbehorende tijdsspanne (`STARTDATE-ENDDATE`).

`STOCK` uit `stock_category` correspondeert met `STOCK` uit `stock_name`, waaruit dan de volledige naam gehaald kan worden.

`CATEGORY` uit `stock_category` correspondeert met `ID` uit `category`, waaruit dan ook de naam gehaald kan worden.

5.52. Stock_codetype (Aandeelmodule)

Column Name	Null?	Type
STOCK	NOT NULL	NUMBER
CODETYPE	NOT NULL	NUMBER
CODE	NOT NULL	VARCHAR2(200)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `stock_codetype` koppelt aan elke `stock` (`STOCK`) het bijbehorende codetype (`CODETYPE`) en de daarbijhorende code (`CODE`) en de desbetreffende periode (`STARTDATE-ENDDATE`) dat deze van kracht is.

Eenzelfde `stock` kan op verschillende manieren genoteerd staan in de koerslijsten (bijvoorbeeld als ISIN-code (International Securities Identification Numbering System), maar ook als SRW-code (Stock Roerende Waarde)) en op deze manier kan al deze info bijgehouden worden in de databank, zonder al te veel gegevens dubbel op te slaan.

`STOCK` uit `stock_codetype` correspondeert met `STOCK` uit `stock_name` waaruit dan de naam gehaald kan worden, `CODETYPE` correspondeert met

ID uit Codetype waaruit dan ook de benaming gehaald kan worden. CODE is dan de exacte code zoals deze terug te vinden is in de koerslijsten (wordt immers opgeslagen als Type VARCHAR2(200)).

5.53. Stock_corporation (Aandeelmodule)

Column Name	Null?	Type
STOCK	NOT NULL	NUMBER
CORPORATION	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `stock_corporation` geeft de koppeling weer tussen een zeker aandeel (STOCK) en een bedrijf (CORPORATION) en de bijbehorende periode (STARTDATE-ENDDATE) dat dit aandeel gekoppeld was aan dit bedrijf.

STOCK uit `Stock_corporation` correspondeert met STOCK uit `Stock_name` waaruit dan de naam (NAME) gehaald kan worden, CORPORATION correspondeert met CORPORATION uit `Corporation_name` waaruit dan de benaming gehaald kan worden.

Een voorbeeldje:

Stock_corporation			
STOCK	CORPORATION	STARTDATE	ENDDATE
10	306	01-JAN-1000	31-DEC-3999
211	619	01-JAN-1000	31-DEC-3999

Stock_name			
STOCK	NAME	STARTDATE	ENDDATE
10	Anvers Gand (CF) (act.prim.) (°1847)	04-MAY-1847	15-JUL-1853
10	Anvers Gand (CF) (act.priv.) (°1847) (1853)	15-JUL-1853	31-JAN-1861
10	Act. priv. : Anvers Gand (CF) (act.priv.) (°1847) (1853) (3])	31-JAN-1861	01-JAN-1878
10	Anvers Gand (CF) (act.priv.) (°1847) (3])	01-JAN-1878	31-DEC-3999
211	Banque Travaux Publiques (des) (société) (°1872)	02-JAN-1872	31-DEC-3999

Corporation_name			
CORPORATION	NAME	STARTDATE	ENDDATE
-----	-----	-----	-----
619	Banque des travaux publics	22-APR-1871	01-JAN-1887
306	Chemins de Fer:Anvers à Gand par St.Nicolas et Lokeren (Cie du CF d+)	25-FEB-1845	31-DEC-3999

5.54. Stock_dividend (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
STOCK	NOT NULL	NUMBER
COUPON		VARCHAR2(250)
TYPE	NOT NULL	NUMBER
EX_COUPON	NOT NULL	DATE
DIVIDEND	NOT NULL	NUMBER
CURRENCY	NOT NULL	NUMBER
YEAR1		VARCHAR2(250)
YEAR2		VARCHAR2(250)
SOURCE		VARCHAR2(2000)
COMMENTS		VARCHAR2(2000)
EURO		NUMBER

De tabel `stock_dividend` koppelt aan elke `stock` (`STOCK`) een dividend (`DIVIDEND`). De coupon (`COUPON`) wordt bijgehouden, het type (`TYPE`), de munteenheid (`CURRENCY`) en nog andere nuttige informatie.

`STOCK` uit `stock_dividend` correspondeert met `STOCK` uit `stock_name` waaruit dan de naam gehaald kan worden, `TYPE` correspondeert met `ID` uit `Dividendtype`.

Deze dividenden stonden in de koersboeken vermeld vanaf 1873. Voor de periode 1832-1872 hebben we beroep gedaan op kranten en diverse monografieën, zodat we nu over een quasi volledige reeks dividenden beschikken.

5.55. Stock_emission (Aandeelmodule)

Column Name	Null?	Type
STOCK	NOT NULL	NUMBER
PRICE	NOT NULL	NUMBER
CURRENCY	NOT NULL	NUMBER
SOURCE		VARCHAR2(2000)
COMMENTS		VARCHAR2(2000)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Stock_emission` geeft voor elke stock (`STOCK`) de emissie (`PRICE`). Er wordt nog andere nuttige info bijgehouden, zoals eventuele opmerkingen (`COMMENTS`), ...

`STOCK` uit `Stock_emission` correspondeert met `STOCK` uit `Stock_name` waaruit dan de naam gehaald kan worden.

`CURRENCY` correspondeert met `ID` uit `Currency`, waaruit dan de volledige naam gehaald kan worden.

De emissieprijzen van nieuwe uitgiften worden verwerkt in de indexberekeningen in "de Grafische Module" (zie Deel II, Modules).

5.56. Stock_name (Aandeelmodule)

Column Name	Null?	Type
STOCK	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(250)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Stock_name` dient voor de naamgeving van de stocks. Bij elke unieke stocknummer (`STOCK`) hoort zijn volledige naam (`NAME`) en de periode dat deze van kracht was (`STARTDATE` - `ENDDATE`).

`STOCK` uit `Stock_name` correspondeert met `ID` uit `Stock`.

`NAME` geeft dan de volledige naam, zoals deze in de koerslijsten terug te vinden is. `STARTDATE` en `ENDDATE` beschrijven de periode waarbinnen deze benaming geldig was.

In de tabel `stock_name` zitten meer tupels dan in de tabel `stock`, wat er dus op duidt dat er ondertussen al een groot aantal naamsveranderingen zijn doorgevoerd.

5.57. Stock_nominal (Aandeelmodule)

Column Name	Null?	Type
-----	-----	----
STOCK	NOT NULL	NUMBER
VALUE		NUMBER
CURRENCY		NUMBER
FRACTION		CHAR(1)
SOURCE		VARCHAR2(2000)
COMMENTS		VARCHAR2(2000)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `stock_nominal` geeft bij elke stock (`STOCK`) zijn nominale waarde (`VALUE`). Ook de juiste munteenheid (`CURRENCY`) wordt bijgehouden, net als de bron van deze gegevens (`SOURCE`) en eventuele opmerkingen (`COMMENTS`) en de corresponderende tijdsspanne (`STARTDATE` - `ENDDATE`).

`FRACTION` wordt gebruikt als er geen "echte" waarde terug te vinden is, maar als de nominale waarde uitgedrukt wordt als een fractie van het kapitaal, bijvoorbeeld 1/10 000 van het kapitaal. Dit is dus als het ware de inverse van het aantal aandelen.

`STOCK` uit `stock_nominal` correspondeert met `STOCK` uit `stock_name` waaruit dan de naam gehaald kan worden.

`CURRENCY` correspondeert met `ID` uit `Currency`, waaruit dan de volledige naam gehaald kan worden.

In de voorbeeldtabel `Stock_nominal` is het commentaar (`COMMENTS`) weggelaten wegens de te grote omvang, net als `FRACTION` en `SOURCE`.

Stock_nominal				
STOCK	VALUE	CURRENCY	STARTDATE	ENDDATE
-----	-----	-----	-----	-----
25	1000	1	24-MAY-1837	10-OCT-1841
25	100	1	10-OCT-1841	01-JAN-1845
25	90	1	01-JAN-1845	01-JAN-1846
25	80	1	01-JAN-1846	31-DEC-3999
36	1000	1	01-JAN-1839	14-JUL-1930
36	0	1	14-JUL-1930	31-DEC-3999
74	1058.2	1	10-JAN-1856	12-OCT-1903
74	25947	1	12-OCT-1903	09-JAN-1905
74	31000	1	09-JAN-1905	31-DEC-1920
74	62000	1	31-DEC-1920	31-DEC-3999

5.58. Stock_quantity (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
STOCK	NOT NULL	NUMBER
TYPE	NOT NULL	NUMBER
VOLUME		NUMBER
VALUE		NUMBER
CURRENCY		NUMBER
SOURCE		VARCHAR2(2000)
COMMENTS		VARCHAR2(2000)
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Stock_quantity` koppelt aan elke stock (`STOCK`) van een zeker type (`TYPE`) de juiste hoeveelheid (`VOLUME`).

Andere belangrijke informatie zoals er uit bovenstaande tabel blijkt (`CURRENCY`, `COMMENTS`,...) wordt ook bijgehouden.

`STOCK` uit `Stock_quantity` correspondeert met `STOCK` uit `Stock_name` waaruit dan de naam gehaald kan worden.

`TYPE` correspondeert met `ID` uit `Stock_type`.

`CURRENCY` correspondeert met `ID` uit `Currency`, waaruit dan ook de volledige naam gehaald kan worden.

We illustreren met een fragment uit de tabel `Stock_quantity` waarin we `SOURCE`, `VALUE` en `COMMENTS` hebben weggelaten wegens te veel informatie. We hebben tevens ook maar enkele gegevens uit de databank over `STOCK 25` genomen.

Stock_quantity				
STOCK	TYPE	VOLUME	STARTDATE	ENDDATE
-----	-----	-----	-----	-----
25	2	5000	24-MAY-1837	01-DEC-1853
25	2	50000	01-DEC-1853	12-FEB-1857
25	2	90000	12-FEB-1857	09-NOV-1859
25	2	112500	09-NOV-1859	30-APR-1860
25	2	103500	30-APR-1860	01-JAN-1864
25	2	112500	01-JAN-1864	31-DEC-1925
25	2	150000	31-DEC-1925	16-NOV-1928

5.59. Stock splits (Aandeelmodule)

Column Name	Null?	Type
-----	-----	-----
STOCK	NOT NULL	NUMBER
SPLITDATE	NOT NULL	DATE
BEFORE	NOT NULL	NUMBER
AFTER	NOT NULL	NUMBER
SOURCE		VARCHAR2(2000)
TOSTOCK		NUMBER

De tabel `stock_splits` is nog een tabel die van zeer groot algemeen belang is voor het SCOB. In deze tabel wordt er voor elke stock (`STOCK`) bijgehouden wanneer deze gesplitst is (`SPLITDATE`), en over hoeveel aandelen het ging "voor" de split (`BEFORE`) en ook "na" de split (`AFTER`). Op deze manier kan er op een eenvoudige manier de zogenaamde "relatieve verhouding" of "ruilverhouding" berekend worden.

Als er bijvoorbeeld 30 000 aandelen voor de split waren, en nog slechts 10 000 erna, is de relatieve verhouding 3/1.

Men houdt ook de bron bij waaruit men de splitsgegevens gehaald heeft (`SOURCE`) en de stock waar "in" alles gesplitst is (`TOSTOCK`). Dit correspondeert met `STOCK` uit `Stock_name` en met `STOCK` in de tabel `Stock_splits` zelf. Op dit moment kan een stock enkel in stukken van zichzelf splitsen. Nog beter zou zijn als dat ook zou kunnen in andere stocks.

Om deze splits eenvoudig te berekenen is er een module "MakeStockInfo" geschreven (waarover meer in **Deel II, Modules**, waarin ook de andere toepassingen van deze module worden beschreven) waarin alle relevante gegevens per stock in chronologische volgorde berekend worden.

5.60. Stock_type (Aandeelmodule)

Column Name	Null?	Type
STOCK	NOT NULL	NUMBER
TYPE	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE

De tabel `Stock_type` geeft voor elke stock (`STOCK`) zijn bijbehorende type (`TYPE`) en de periode dat dit van kracht is (`STARTDATE` - `ENDDATE`).

`STOCK` uit `Stock_type` correspondeert met `STOCK` uit `Stock_name` waaruit dan de naam gehaald kan worden, `TYPE` correspondeert met `ID` uit `Stocktype`.

5.61. Subarchivenummer (Documentmodule)

Column Name	Null?	Type
ID	NOT NULL	NUMBER
ARCHIVENUMBER		NUMBER
NAME		VARCHAR2(200)
PARENT		NUMBER
STARTDATE		DATE
ENDDATE		DATE

De tabel `subarchivenummer` is een tabel die gebruikt wordt in het licht van de classificatie van de documenten die het SCOB nu bezit.

Elke "gebeurtenis" krijgt zijn eigen identificatienummer (`ID`), dat gekoppeld is aan de volledige naam (`NAME`) van de gebeurtenis die verschijnt in "de Documentmodule". De "sub" in de naam

Subarchivenummer verwijst naar een soort boomstructuur die er tussen de documenten gecreëerd wordt. Er wordt dan ook van een document het "overkoepelende" document bijgehouden (PARENT) en de periode dat dit zo was (STARTDATE - ENDDATE).

We geven een kleine illustratie van de werking van de tabel Subarchivenummer. Hiertoe vertrekken we van de tabel Archiver.

Archiver					
ID	NAME	ARCHIVER	PUBLISHER	STARTDATE	ENDDATE
1	A. 1	1	9898	01-JAN-1980	31-DEC-1995
2	A. 2	1	5680	01-JAN-1904	31-DEC-1989

We bekijken in het bijzonder de 2 archieven met naam A. 1 en A. 2, die thuishoren onder de ARCHIVER met nummer 1, de Beurs van Brussel (zie tabel Archiver (5.2.)).

Deze corresponderen met de uitgevende bedrijven (PUBLISHER) 9898 en 5680, waarvan we in Corporation_name de volledige benaming kunnen terugvinden.

Corporation_name			
CORPORATION	NAME	STARTDATE	ENDDATE
5680	Ateliers de Constructions Électriques de Charleroi	07-JUL-1904	31-DEC-3999
9898	ABAY S.A.	01-JAN-1880	31-DEC-3999

In Archiver wordt A. 1 voorgesteld d.m.v. ID 1 en A. 2 d.m.v. ID 2. In de tabel Subarchivenummer wordt er nu een onderverdeling aan elk van deze toegekend.

Zo zien we dat A. 1 (ARCH.NUM = ID = 1) vijf "onderverdelingen" heeft (de eerste vijf afgedrukte tupels (ID 1 t.e.m. 5)), en A. 2 (ARCH.NUM = ID = 2) negen (ID 5 t.e.m. 14).

Subarchivenumber					
ID	ARCH.NUM	NAME	PARENT	STARTDATE	ENDDATE
---	-----	-----	-----	-----	-----
1	1	Knipseldossiers	0	01-OCT-1980	31-DEC-1988
2	1	Map 1	1	01-OCT-1980	31-DEC-1988
3	1	Bedrijfsdossiers	0	01-JAN-1000	31-DEC-3999
4	1	Briefwisseling	0	01-OCT-1981	30-JUN-1983
5	1	Map 2	1	01-JAN-1989	31-DEC-1995
6	2	Knipseldossiers	0	01-APR-1943	31-DEC-1955
7	2	Map 1	6	01-APR-1943	31-DEC-1955
8	2	Bedrijfsdossiers	0	01-JAN-1904	31-DEC-1970
9	2	Briefwisseling	0	01-MAR-1976	31-MAY-1982
10	2	Map 2	6	01-FEB-1956	31-DEC-1968
11	2	Map 3	6	01-FEB-1969	31-DEC-1976
12	2	Map 4	6	01-MAR-1977	31-DEC-1982
13	2	Map 5	6	01-JAN-1983	30-JUN-1986
14	2	Map 6	6	01-JUL-1986	31-JUL-1989

Bij deze "onderverdelingen" is het zeer belangrijk te kijken naar PARENT. Als PARENT 0 is, betekent dit dat deze onderverdeling rechtstreeks onder zijn "archiver" hangt. Is deze onderverdeling een andere ID, dan wordt er een "subboom" gecreëerd. Zo zien we dat er onder "doos" A. 1 (ARCH.NUM 1) rechtsreeks de "mappen" Knipseldossiers (ID 1), Bedrijfsdossiers (ID 3) en Briefwisseling (ID 4) hangen. Zij hebben immers alledrie PARENT 0. Map 1 (ID 2) en Map 2 (ID 5) hebben echter PARENT 1, wat er op wijst dat zij "submappen" zijn van de "map" met ID 1, Knipseldossiers. Analoog zien we dat "doos" A. 2 (ARCH.NUM 2) onderverdeeld is in de "mappen" Knipseldossiers, Bedrijfsdossiers en Briefwisseling (ID 6, 8 en 9) en dat de "map" Knipseldossiers (ID 6) nog is onderverdeeld in de "submappen" Map 1 t.e.m. 6 (ID 7, 10, 11, 12, 13 en 14).

Aan elk van deze "subarchivenumbers" (ID's uit Subarchivenumber), dus "mappen" en eventuele "submappen" kunnen nu echte documenten worden toegewezen. Dit gebeurt d.m.v. de tabel Document_SAN. Deze koppelt aan elk subarchivenumber (SAN) een document (DOCUMENT).

Document_SAN	
DOCUMENT	SAN
-----	---
15	2
16	2
17	2
18	2
19	2
20	2
21	2
22	2
23	2
24	2

Zo zien we dat aan SAN 2 de documenten 15 t.e.m. 24 worden toegewezen.

SAN 2 staat voor ID 2 in Subarchivenummer, dus voor Map 1 (onder Knipseldossiers onder A. 1).

De exacte benaming van de toegewezen documenten (want hier wordt er met unieke identificatienummers gewerkt) is terug te vinden in de tabel Document_title. We hebben er hier enkelen opgezocht.

Document_title	
DOCUMENT	NAME
-----	-----
17	Uittreksel uit het jaarverslag van Electrorail
19	Uittreksel uit het jaarverslag van S.N.I.
22	Jaarverslag

Op deze manier kan er dus een boomstructuur gecreëerd worden waarin alle documenten kunnen geplaatst worden.

Het is de bedoeling van "de Documentmodule" (zie **Deel II, Modules**) dat deze structuur op een eenvoudige manier in de databank ingevoerd en uitgebreid kan worden. In tweede instantie dient "de Documentmodule" voor het reflecteren van deze structuur en moet ze ondervraging op de onderliggende mappen en documenten mogelijk maken.

5.62. Valuta (Aandeelmodule)

Column Name	Null?	Type
CURRENCY	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE
FACTOR		NUMBER
COMMENTS		VARCHAR2(250)
SOURCE		VARCHAR2(250)
NAAM	NOT NULL	VARCHAR2(10)

De tabel `Valuta` is een verzameltabel waarin alle valuta die er ooit gebruikt kunnen zijn en kunnen worden in zijn opgenomen.

Elke munteenheid heeft zijn eigen persoonlijke identificatienummer gekregen (`CURRENCY`), en de datum van waarop hij voor het eerst "in omloop" is gebracht (`STARTDATE`), zijnde van wanneer hij voor de eerste keer in de koerslijsten is opgemerkt en waarop hij "uit de handel" is genomen (`ENDDATE`), zoals (nog actueel) de Belgische Frank, die plaats heeft moeten ruimen voor de EURO. Naast zijn volledige naam (`NAAM`) en eventueel commentaar (`COMMENTS`) en de bron (`SOURCE`) wordt ook de factor (`FACTOR`) bijgehouden voor de omrekening naar Belgische Frank.

Voor het optimale gebruik van deze tabel moet er nog een "ValutaModule" geschreven worden.

6. Logische samenhang van de tabellen.

In 5. hebben we een gedetailleerd overzicht gegeven van de individuele tabellen die er voorkomen in de databank. Deze opsomming is niet de ideale manier om een globaal beeld te krijgen van de databank. Daarom geven we nu een overzicht van **de logische samenhang** van de tabellen. Op deze manier krijgt men wel een duidelijk overzicht van de werking van de SCOB-databank en de interne interactie hierin tussen de verschillende tabellen.

De meest logische manier van groeperen van de tabellen is **per module** waarin ze hoofdzakelijk gebruikt worden. Meer uitleg over de modules zelf is terug te vinden in **Deel II, Modules**.

We beginnen telkens met een opsomming van de tabellen die er hoofdzakelijk per module gebruikt worden. Natuurlijk is er een zekere overlapping en kunnen bepaalde tabellen in meerdere modules gebruikt worden. In dit geval plaatsen we ze de tabel bij de meest typerende module. Daarna verklaren we wat de exacte werking is van deze tabellen en hoe ze onderling verbonden zijn. Voor meer gedetailleerde informatie verwijzen we terug naar 5. waarin de afzonderlijke tabellen beschreven staan.

6.1. De Aandeelmodule

- 5.5. Category
- 5.6. Codetype
- 5.24. Currency
- 5.25. Dividendtype
- 5.32. Notation
- 5.33. Notation_price
- 5.34. Notation_tradedquantity
- 5.38. Quantitytype
- 5.40. Sector
- 5.41. Sector_child
- 5.42. Sector_name
- 5.43. Sector_next
- 5.45. Sharetype
- 5.46. Stock
- 5.47. Stockexchange
- 5.48. Stockexchange_child
- 5.49. Stockexchange_name
- 5.50. Stocktype
- 5.51. Stock_category
- 5.52. Stock_codetype
- 5.53. Stock_corporation
- 5.54. Stock_dividend
- 5.55. Stock_emission
- 5.56. Stock_name
- 5.57. Stock_nominal
- 5.58. Stock_quantity
- 5.59. Stock_splits
- 5.60. Stock_type
- 5.62. Valuta

De Aandeelmodule is de module die zo veel mogelijk informatie groepeert over aandelen, obligaties, dividenden, ...

De informatie die er ter beschikking is van het SCOB, is in de databank gerangschikt per beurs. Hiervoor worden de tabellen `Stockexchange`, `Stockexchange_child` en `Stockexchange_name` gebruikt. In `Stockexchange` worden de beurzen geïntialiseerd door aan elke beurs een uniek nummer toe te wijzen. In `Stockexchange_name` wordt aan elk van die nummers de naam van de desbetreffende beurs gekoppeld. Met `Stockexchange_child` kan er dan aan elke beurs zijn eerste sector uit de koersboekstructuur gekoppeld worden. De sectoren die op hetzelfde niveau staan, worden hieraan gekoppeld met `Sector_next` (zie vorige alinea).

Er wordt een volledige boomstructuur in de databank bewaard van alle sectoren zoals deze terug te vinden zijn in de koerslijsten. Elke wijziging in de koersboekstructuur wordt bewaard op de datum dat deze plaatsvond. Deze structuur wordt zichtbaar gemaakt in het linkerframe van de Aandeelmodule. De tabellen die hiervoor gebruikt worden zijn `Sector`, `Sector_child`, `Sector_name` en `Sector_next`. In `Sector` worden alle sectoren geïntialiseerd, dus "gemaakt" door ze een uniek nummer te geven. In `Sector_name` krijgen ze dan hun naam toegewezen aan dit nummer. Met `Sector_child` kan er een eerste subsector gemaakt worden waaraan dan op hetzelfde niveau nog andere sectoren gekoppeld kunnen worden met `Sector_next`.

Zoals reeds eerder vermeld draait deze module hoofdzakelijk rond aandelen en obligaties. Hiervoor wordt er gebruik gemaakt van de tabellen `Stock`, `Stocktype`, `Category`, `Codetype`, `Corporation`, `Dividendtype`, `Quantitytype`, `Stock_category`, `Stock_codetype`, `Stock_corporation`, `Stock_dividend`, `Stock_emission`, `Stock_name`, `Stock_nominal`, `Stock_quantity`, `Stock_splits` en `Stock_type`. De enkelvoudige tabel `Stock` initialiseert alle stocks door ze intern een uniek identificatienummer te geven. De enkelvoudige tabel `Stocktype` definieert alle types van stocks die er kunnen voorkomen in de koerslijsten. De enkelvoudige tabel `Category` specificeert alle mogelijk categorieën door ze een nummer toe te wijzen. De enkelvoudige tabel `Codetype` definieert de verschillende soorten van codes die er in koerslijsten kunnen voorkomen. De enkelvoudige tabel `Corporation` initialiseert elk bedrijf door het een uniek nummer te geven. De enkelvoudige tabel `Dividendtype` specificeert alle mogelijke types van

dividenden die er kunnen voorkomen. De enkelvoudige tabel `Quantitytype` specificeert de verschillende soorten hoeveelheden die er kunnen voorkomen. De samengestelde tabellen (de tabellen met de “_” in) gebruiken de info uit de enkelvoudige tabellen. Zo wordt er in `Stock_name` aan elke stocknummer uit `Stock` de juiste naam gelinkt. In `Stock_category` wordt elke stock gekoppeld aan de juiste categorie die terug te vinden is in `Category`. `Stock_codetype` koppelt dan weer aan elke stock zijn codetype (uit `Codetype`) en de corresponderende code. `Stock_corporation` linkt elke stock aan het bedrijf (uit `Corporation`) dat deze uitgeeft. Meer informatie over de verschillende bedrijven is terug te vinden in **6.2**. `Stock_dividend` koppelt aan elke stock zijn dividend (uit `Dividendtype`), `Stock_emission` doet hetzelfde met emissieprijsen. `Stock_nominal` linkt aan elke stock zijn nominale waarde, `Stock_quantity` doet dit met hoeveelheden (uit `Quantitytype`), `Stock_splits` met stock splits en `Stock_type` met het type dat uit `Stocktype` komt.

De koersen en de verhandelde hoeveelheden van de verschillende aandelen worden bijgehouden in de tabellen `Notation`, `Notation_price` en `Notation_tradedquantity`. De koersen worden nooit rechtstreeks aan een aandeel gekoppeld, maar krijgen een eigen notatienummer in `Notation`. De exacte koerswaarden worden in `Notation_price` gekoppeld aan dit nummer, net zoals de hoeveelheden gekoppeld worden aan dit nummer in `Notation_tradedquantity`.

Er blijven nu nog de tabellen `Currency`, `Sharetype` en `Valuta` over. `Currency` geeft aan elke munteenheid een uniek nummer. `Valuta` is een verzameltabel waarin alle valuta die er ooit gebruikt kunnen zijn, zijn opgenomen. `Sharetype` geeft een unieke identificatiesleutel aan elk type van activa, zoals aandeel, obligatie, ...

6.2. De Bedrijfsmodule

- 5.3. Bank
- 5.4. Bookkeepingitem
- 5.7. Corporation
- 5.8. Corporationaction
- 5.9. Corporationset
- 5.10. Corporation_adminlocation
- 5.11. Corporation_bank
- 5.12. Corporation_bookkeeping

- 5.13. Corporation_bookkeepingitem
- 5.14. Corporation_capitalchanges
- 5.15. Corporation_evolution
- 5.16. Corporation_information
- 5.17. Corporation_location
- 5.18. Corporation_name
- 5.19. Corporation_resultitems
- 5.20. Corporation_results
- 5.21. Corporation_sociallocation
- 5.22. Corporation_stockholders
- 5.23. Corporation_stocks
- 5.30. Evolutionrole
- 5.31. Job
- 5.35. Person
- 5.36. Person_function
- 5.37. Person_location

De Bedrijfsmodule groepeerd zo veel mogelijk informatie over de verschillende bedrijven die er in de koersboeken genoteerd staan en waarover het SCOB informatie heeft.

In de tabel Corporation wordt er aan elk bedrijf een uniek identificatienummer gegeven. De koppeling tussen de nummers en de bedrijfsnamen gebeurt dan in Corporation_name. De andere "samengestelde" tabellen koppelen extra informatie over de bedrijven aan het bedrijf in kwestie. Zo beschrijft Corporation_adminlocation de locatie van de hoofdzetel van het bedrijf. Het nummer van de bank van een bedrijf is terug te vinden in Corporation_bank en de benaming van de bank die correspondeert met dat nummer in Bank. In de tabel Corporation_bookkeepingitem worden de nummers bijgehouden van de items die beschreven staan in de tabel Bookeepingitem. Deze twee tabellen beschrijven de boekhouding van een bedrijf. Verder wordt er per bedrijf info bijgehouden over de kapitaalsveranderingen in de tabel Corporation_capitalchanges. Voor elk bedrijf kunnen er extra opmerkingen opgeslagen worden in Corporation_information. De locatie van de maatschappelijke zetel van een bedrijf wordt opgeslagen in Corporation_location, die van de administratieve zetel in Corporation_sociallocation. Of een bedrijf winst heeft gemaakt of verlies is op te zoeken in Corporation_resultitems. Van een bedrijf wordt er ook bijgehouden wie de aandeelhouders zijn (in Corporation_stockholders) en welke aandelen er zijn uitgegeven (in

Corporation_stocks). De evoluties die een bedrijf kan doormaken worden beschreven in `Evolutionrole`.

In een bedrijf worden natuurlijk een aantal mensen tewerkgesteld. Deze gegevens worden opgeslagen in de tabel `Person`. De mogelijke beroepen die iemand kan uitoefenen worden opgeslagen in `Job`. Welke persoon nu welke functie uitoefent is terug te vinden in de tabel `Person_function`, en waar elke persoon woont is terug te vinden in `Person_location`.

6.3. De Documentmodule

- 5.1. Archivenumber
- 5.2. Archiver
- 5.26. Document
- 5.27. Documenttype
- 5.28. Document_san
- 5.29. Document_title
- 5.44. Sessionpart
- 5.61. Subarchivenumber

De Documentmodule is de module die zo veel mogelijk gegevens verzamelt over alle documenten en archieven die het SCOB ter beschikking heeft.

Wat betreft de uitgevers van deze documenten zijn er in de databank vier tabellen terug te vinden. In `Archivenumber` wordt aan elk archief dat het SCOB bezit een nummer toegekend. Deze archieven worden hoofdzakelijk bewaard in het lokaal van SCOB. `Archiver` beschrijft de persoon of de instelling die deze archieven bijgehouden hebben voordat ze in handen kwamen van SCOB. De tabellen `Sessionpart` en `Subarchivenumber` zijn tabellen die gebruikt worden om een boomstructuur in de documenten te maken. Zo kan elke "doos" immers bestaan uit "mappen" die op hun beurt "knipsels" kunnen bevatten. Meer uitleg over het exact functioneren van deze boomstructuur is terug te vinden bij de beschrijving van de afzonderlijke tabellen.

Ook van de documenten zelf wordt er een heleboel informatie bijgehouden. In `Document` krijgt elk document een uniek nummer toegewezen. De koppeling van deze nummers met de bijbehorende naam gebeurt in `Document_title`. In de tabel `Documenttype` worden de verschillende types van documenten gespecificeerd. `Document_san` koppelt aan elk document ("doos") zijn subdocumenten ("map").

Deel II

De Modules

1. Inleiding

In het volgende deel behandelen we de verschillende “modules” en belangrijke documenten die er geschreven zijn “op” de SCOB-databank en die dagelijks gebruikt worden voor de **invoer** en **verwerking** van de data uit de koerslijsten.

We beschrijven achtereenvolgens

- De Aandeelmodule
- De Bedrijfsmodule
- De Outputmodule
- De Documentmodule
- MakeStockInfo
- MakeMatrix
- De Logboeken

Van elke “module” geven we een gedetailleerde beschrijving, gecombineerd met de nodige illustraties. Zo krijgt de lezer een concreet beeld van de gebruiksvriendelijkheid en het nut en wordt alles concreter gemaakt.

2. De Aandeelmodule

De "Aandeelmodule" is toch wel de module die het vaakst gebruikt wordt op het SCOB. Deze JAVA-module is geschreven met de bedoeling om op een eenvoudige, overzichtelijke en snelle manier **gegevens** uit de koerslijsten **in de databank te kunnen ingeven**. Anderstalige mensen en mensen die bijna nooit met een computer in aanraking zijn gekomen, kunnen er snel mee werken, mits een kleine initialisatie, die dan ook nog hoofdzakelijk gaat over de uitzonderingen die je kan tegenkomen in de koerslijsten, en dus niet over het gebruik van de module zelf. Voor een geroutineerd computergebruiker spreekt de module voor zichzelf.

We zullen nu in het kort uitleggen hoe de module eruitziet en gebruikt wordt.

Als de module opstart, verschijnt er een schermpje waarin de datum moet ingegeven worden van de dag waarop men wenst gegevens in te voeren of te raadplegen.

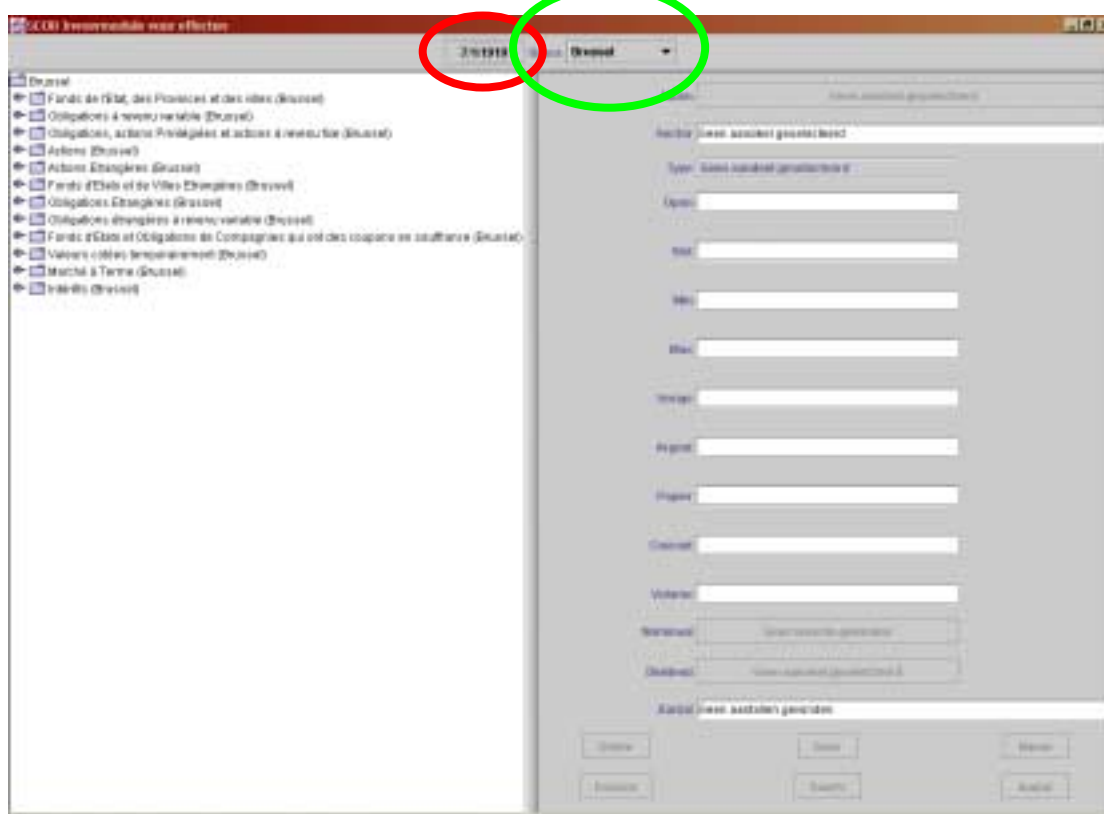


The image shows a small dialog box titled "Datum Invoer". On the left side, there are three labels: "dag", "maand", and "jaar", each corresponding to an empty text input field on the right. At the bottom center of the dialog box is an "Ok" button. The window has a standard title bar with a close button (X) in the top right corner.

Men vult de datum in, klikt op **ok** en de module start vanzelf op.

Als voorbeeld hebben we hier gewerkt met 7 januari 1919.

Men krijgt dan volgend scherm te zien.



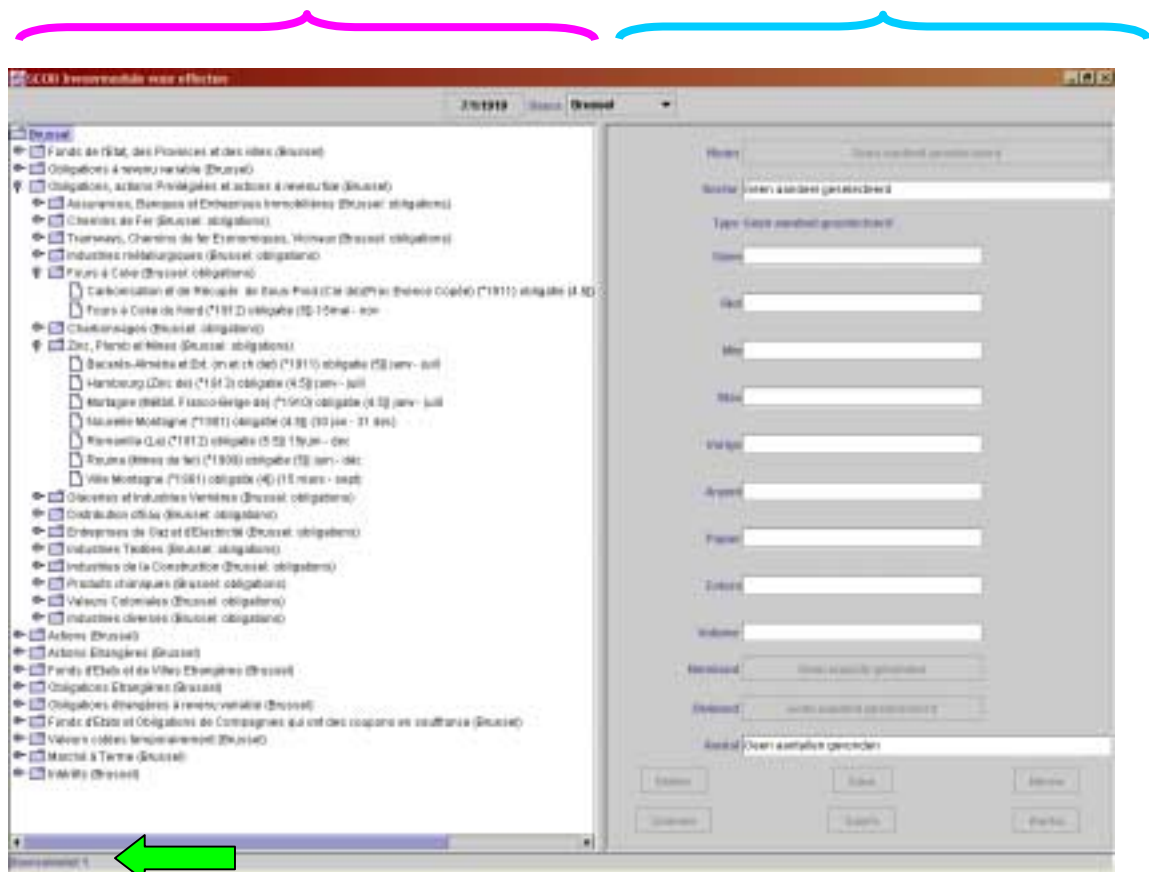
Als voorbeelddatum is hier dus 7 januari 1919 gekozen. Deze datum wordt als geheugensteuntje nog eens centraal weergegeven in de bovenbalk. Indien gewenst kan er hier ook voor een andere datum gekozen worden door op het balkje te klikken. Er verschijnt bijna hetzelfde schermpje als bij het opstarten, enkel een CANCEL-knop is er nieuw bijgevoegd.

Daarnaast staat er ook over welke beurs het gaat. Op dit ogenblik is er enkel gewerkt met gegevens van de Beurs van Brussel. Men kan wel andere beurzen kiezen (bij het "openklappen" van de keuzebalk, zoals te zien is op de illustratie), dus in de toekomst kan er ook begonnen worden aan de invoer van gegevens voor andere beurzen.



Het hoofdscherm zelf is verdeeld in **twee** grote blokken en statuslijn.

I. Het **linkerblok** is een weergave in “boomstructuur” van de structuur van de koerslijsten op de gekozen datum. Bij het opstarten wordt enkel de hoofdboom weergegeven, maar met enkele eenvoudige toetsbewegingen of een simpele muisklik, kan men de “onderliggende” structuur zichtbaar maken. Lange tekst kan gelezen worden door gebruik te maken van de scroll-balk beneden, maar ook door het scheidinglijn tussen de twee helften met de muis te verslepen.



Deze “koersboekboomstructuur” die verschijnt in de linkerhelft van het scherm wordt door de module uit de databank gehaald. Er zijn hiervoor een aantal tabellen ontworpen (waaronder `Sector_child` en `Sector_next`)(zie **Deel I, De Databank**) die zodanig met elkaar verbonden zijn dat ze een soort “nesting” veroorzaken.

Deze koersboekstructuur laat zowel een *handmatige* input van gegevens toe als een *elektronische* input en koppelt de ingevoerde gegevens aan het juiste aandeel. Voor de handmatige input heeft de gebruiker de boomstructuur uit het linkerscherm nodig als reflectie van de koerslijsten die hij naast zich heeft liggen en van waaruit hij de gegevens overneemt.

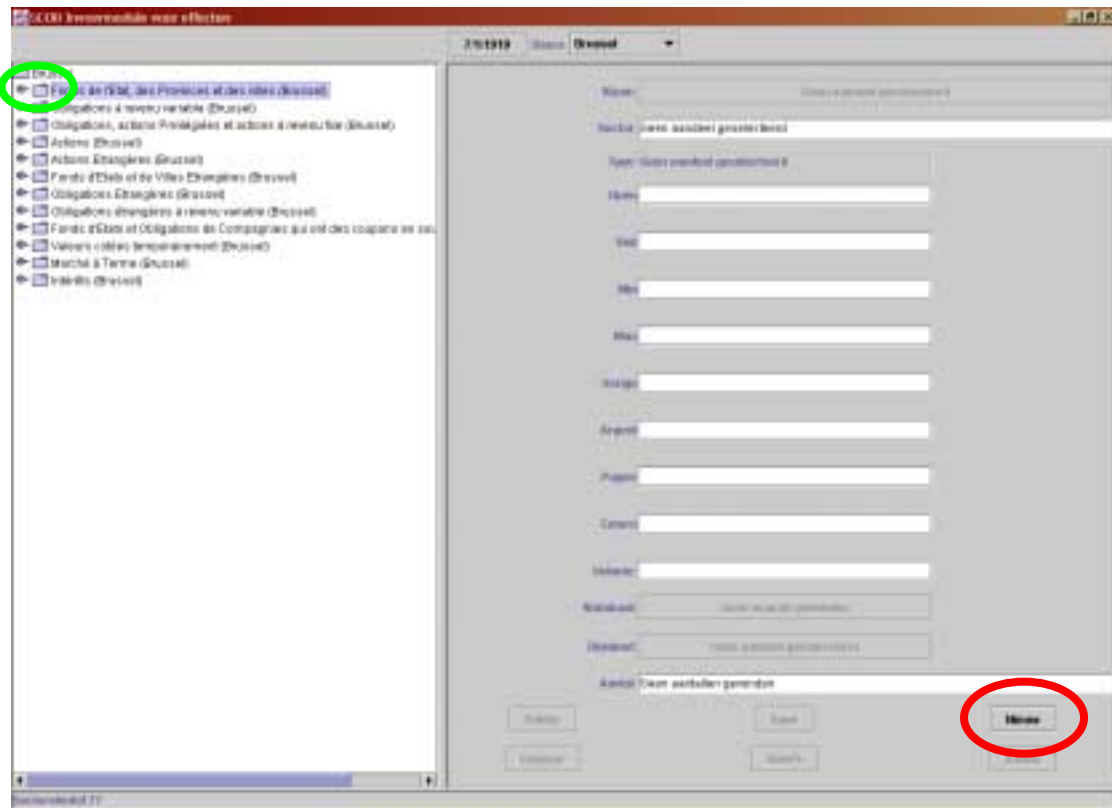
Om de databank dus correct op te stellen was het bijgevolg precies uitzoeken van de koersboekstructuur noodzakelijk. Deze omvat alle markten en sectoren waaronder de aandelen en obligaties staan en wijzigt continu doorheen de tijd.

De structuur die men op dit moment kan terugvinden in de linkerhelft van de schermindeling is een reflectie van de structuur die men terug kan vinden in de koerslijsten uit de koersboeken van de Beurs van Brussel en in een later stadium ook uit de Financieel Economische Tijd. Elke dag is manueel nagekeken op wijzigingen en telkens er een kleine of een gigantisch grote (de structuur is meerdere malen totaal gewijzigd) koersboekwijziging optrad, is deze onmiddellijk op de juiste datum in de databank doorgevoerd. Dit zeer nauwgezette en intensieve werkje heeft vele weken in beslag genomen maar op dit moment zit de hele structuur in de databank, tot op een maand van de datum van vandaag. De structuur wordt maandelijks nagekeken en bijgewerkt. De hele koersboekevoluntie heeft ons een uitgebreid naslagwerk vol nuttige informatie opgeleverd over alle koersboekwijzigingen die er tot nu toe hebben plaatsgegrepen.

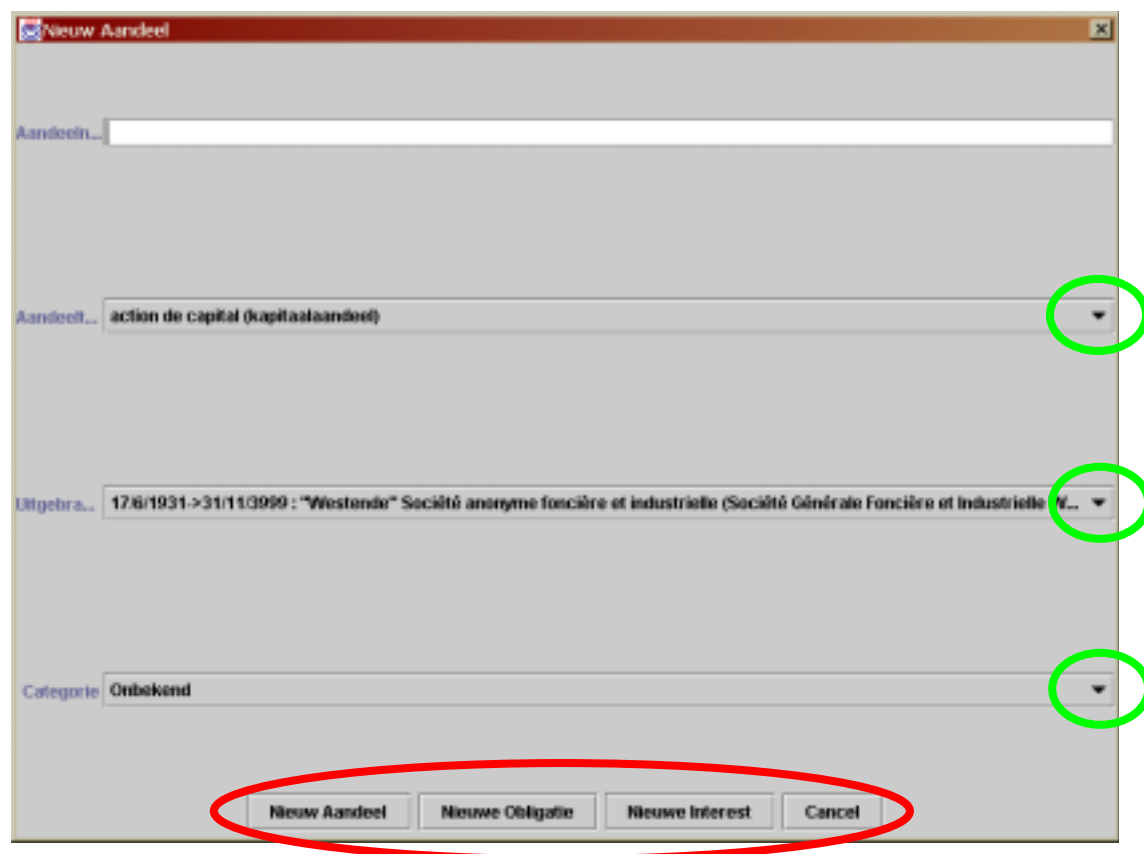
II. Het **rechterblok** kan verschillende functies hebben, afhankelijk van op welke lijn tekst in het linkerblok de cursor staat.

II. a. Staat men op een **sectortitel** op het allerlaagste niveau van de sectoren en subsectoren, dan verschijnt er in het rechterscherm alleen maar de mogelijkheid om onder deze sector een nieuw "ding" aan te maken, een *aandeel* of een *obligatie* of een *intrest*. Deze laagste sectortitel wordt aangeduid d.m.v. een "mapje"-icoon (zie illustratie). Men krijgt immers in het linkerscherm een boomstructuur waarbij er onder sectoren (aangeduid met "mapje"-icoon) nog subsectoren en eventueel subsubsectoren kunnen hangen (of er zelfs nog dieper gegaan kan worden). Het is enkel op het allerlaagste niveau van de "mapjes" in deze boomstructuur dat er een "blad" voor een aandeel of obligatie of intrest kan bijgevoegd worden.

In het voorbeeld staat de cursor op *Fonds de l'état, des Provinces et des Villes*. Hieronder zijn er geen subsectoren meer, dus deze sector staat op het laagste niveau in de boomstructuur. Hieronder kan er dus enkel iets "nieuws" ingevoerd worden. In de rechterbenedenhoek is er alleen de mogelijkheid *Nieuw actief* gemaakt.



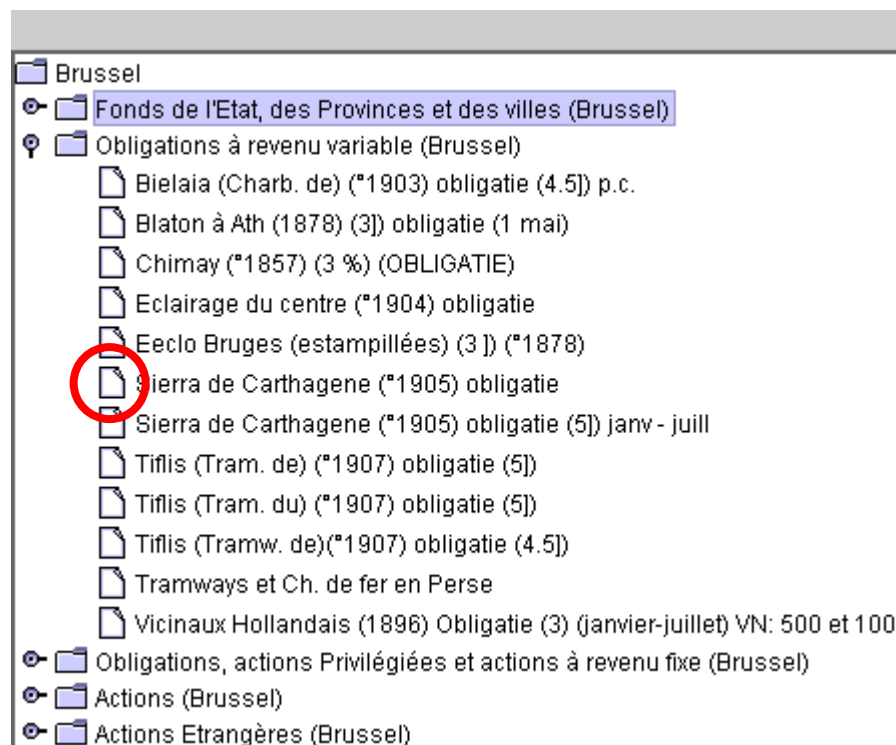
Als men op Nieuw klikt, verschijnt er volgend scherm.



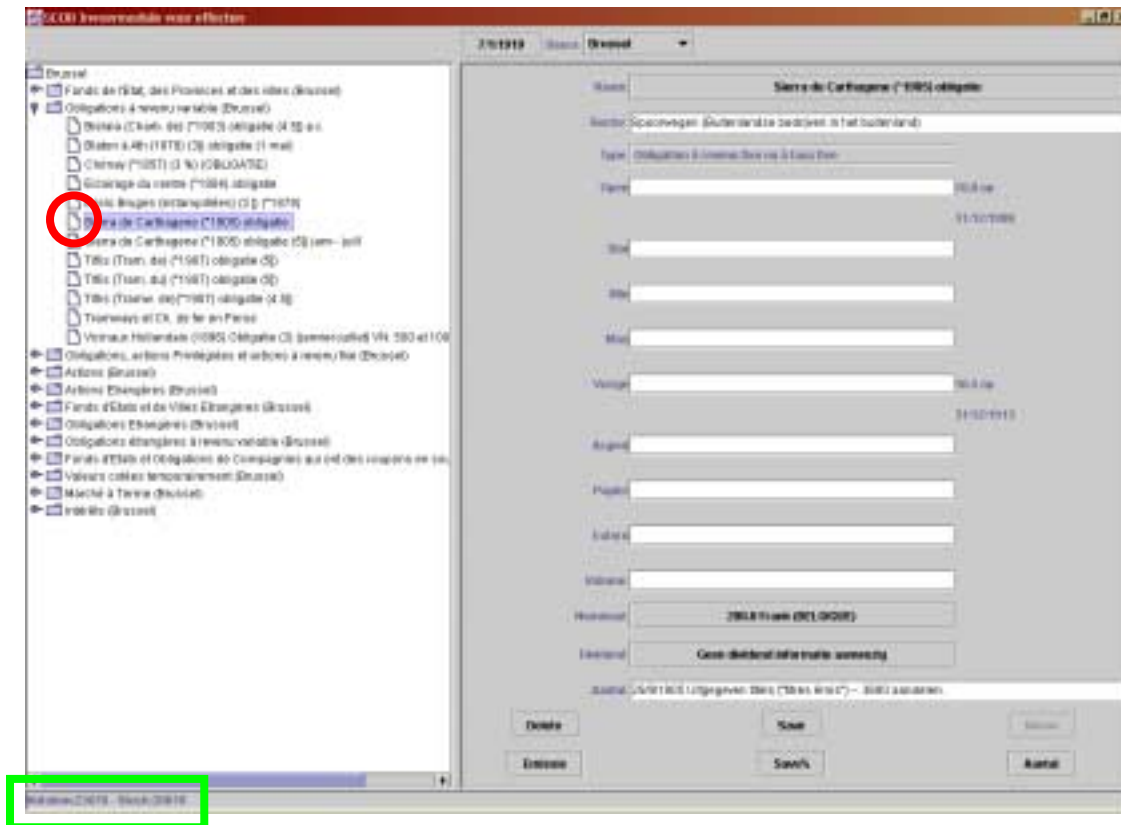
Hierin kan men dan de naam ingeven, en nog enkele andere belangrijke gegevens, zoals de titel, wanneer het "iets" is uitgebracht en de categorie. Elk van deze drie kunnen gekozen worden uit een uitgebreide keuzelijst die uitklapbaar is d.m.v. het pijltje (zie illustratie). Uiteindelijk moet men, om de invoer definitief te maken, kiezen of het nieuwe "ding" wat er gaat bijgemaakt worden een *aandeel* is, of een *obligatie* of een *intrest*. Dit doet men door te klikken op de gewenste knop (zie illustratie). Er wordt op dit moment ook gewerkt aan de mogelijkheid om een *droit* (*recht*) in te geven. Heeft men de gewenste keuze met een muisklik bevestigd, dan is het bijmaken definitief en staan de gegevens in de databank.

Het bijgemaakte "iets" verschijnt dan ook in de boomstructuur *onder* de sectortitel. Op deze manier verkrijgt men de subboomstructuur. De titels met een "blad" voor zijn zulke bijgemaakte aandelen of obligaties, terwijl de "mapjes" de hoofdstructuur aangeven.

Als voorbeeld kijken we hier naar de sector Obligations à revenu variable. Hieronder zijn reeds een heleboel obligaties bijgemaakt, zoals bijvoorbeeld Sierra de Carthagene (°1905) obligatie.



II. b. Staat men echter met het balkje op zo'n gemaakt **aandeel** of **obligatie**, dan verschijnt er op de rechterhelft van het scherm de informatie over dat betreffende aandeel of obligatie.



Als voorbeeld zijn we op de net beschouwde obligatie Sierra de Carthagene (°1905) obligatie gaan staan, en het scherm toont ons de informatie die er over deze obligatie beschikbaar is. Zo komt deze obligatie uit de Spoorwegsector en heeft ze een nominale waarde van 200.0 BEF. Ook het aantal kunnen we uit het veldje aflezen.

III. Tenslotte is er nog **de statusbalk**, beneden aan het scherm, die de interne databankcode weergeeft van wat er nu "in beeld" is. Als we op een sector staan, wordt de sectorcode weergegeven, en als we op een aandeel of obligatie staan, wordt hiervan de interne notatiecode weergegeven en het stocknummer.

We illustreren dit met twee voorbeelden van schermen.



IV. We gaan nu nog wat dieper in op **het aanmaken van de aandelen zelf**. Vooraleer immers de operators (voor SCOB zijn dat hoofdzakelijk jobstudenten geweest) koersen uit de lijsten en boeken kunnen invoeren via hun terminal (PC), moeten de aandelen en obligaties in de databank "bestaan". Hiertoe moesten ze eerst een voor een opgezocht worden in de koersboeken en handmatig aangemaakt worden in de databank volgens het principe beschreven op de vorige bladzijden. Het opzoeken in de koersboeken is gebeurd door jaar-op-jaar alle aandelen en obligaties te vergelijken. Daarna werden ze een voor een aangemaakt. Hiertoe moesten wel eerst de bedrijven geïdentificeerd worden. Aandelen worden immers uitgegeven door bedrijven, obligaties door de staat of door gemeenten of bedrijven. We zijn hier dan ook begonnen met het opzoeken van de bedrijven en dit te combineren met ineens ook andere beschikbare info aan de bedrijven te koppelen (zie ook "de Bedrijfsmodule"). Zodoende vormen de bedrijven als het ware de kern.

Voor elk aandeel dat er door SCOB verwerkt is geworden, is er dus onderzocht van welk bedrijf dit komt. De naam van een aandeel in de koersboeken is immers dikwijls een verkorte benaming en de oorsprong is bijgevolg niet altijd duidelijk. Enkel door gedetailleerd zoekwerk is het mogelijk om elk aandeel uniek te bepalen en de opvolging van de aandelen doorheen de tijd te volgen, samen met het type van operatie (fusie, opslorping,...), iets wat niet automatisch uit de koersboeken blijkt. Al deze informatie is gebundeld in het "Logboek Aandelen", waarop dieper wordt ingegaan verder in **Deel II (8.)**.

De aandelen worden hoofdzakelijk aangemaakt via "de Aandelenmodule" (die hier haar naam aan ontleent). Het fysiek aanmaken van aandelen via deze gespecialiseerde module gaat zeer vlot en werkt overzichtelijk. We hebben hierboven reeds de schermen laten zien die hiervoor gebruikt worden. Het is in het **rechterscherm** van de Aandelenmodule (zie **11.**) dat de koersen ingevoerd moeten worden. Je gaat met de cursor op het aandeel staan waarvan je de koersen wil ingeven en het rechterscherm geeft een invulformulier hiervoor. Al naargelang welke koeren terug te vinden zijn in de koerslijsten worden deze gegevens dan ingevoerd in de velden *Open*, *Slot*, *Min*, *Max*, *Vorige*, *Argent*, *Papier* of *Extern*. Om de koersen fysiek op te slaan moet er nu nog op *Save* geklikt worden. Het resultaat van het invullen van deze velden is het toewijzen van een unieke aandelsleutel en noterings sleutel in de databank. Om een aandeel af te sluiten op de dag dat men aan het werken is, klikt men op *Delete*. Op deze manier verschijnt het aandeel vanaf de volgende dag niet meer in de lijst. In de databank is door de klik op *Delete* het tupel van het desbetreffende aandeel in de *Notation* tabel immers op die datum afgesloten. Fysiek in de databank komt dit neer op *ENDDATE* in *Notation* die die datum is geworden (zie **5.32.**). De aantallen kunnen ingevoerd worden door op de knop *Aantal* te klikken. Er opent zich dan een hulpschermje waarin de gegevens kunnen ingevoerd worden. Door op *Emissie* te klikken kan ook deze informatie in de databank gestopt worden.

Dezelfde procedure kan gebruikt worden voor obligaties. Het is trouwens hiervoor dat de knop *Save%* dient. Soms staan er in de koerslijsten bij obligaties geen gewone waarden maar wel procenten. Om dit mee te geven in de databank moet er op *Save%* gedrukt worden i.p.v. op *Save* bij het bevestigen (opslaan) van de ingevulde gegevens.

Wat er precies gebeurt in de databank zullen we nu verder uitwerken voor het aandeel "Nationale Bank".

Het aandeel "Nationale Bank" (dit is "onze" benaming, de echte benaming is in het Frans, zoals later zal blijken) wordt eerst aangemaakt in de tabel *Stock*. Hierin wordt er een unieke sleutel toegewezen. In de tabel *Stock_name* wordt dan de naam van dat aandeel aan die sleutel gekoppeld, net als de officiële startdatum. Dit is de exacte datum dat het aandeel voor het eerst in de koerslijsten is terug te vinden. De einddatum wordt standaard op 31/12/3999 gezet, totdat men elders (en gewoonlijk pas in de loop van de tijd) een exacte afsluitdatum vindt. Voor het aandeel

“Nationale Bank” is de startdatum 16/01/1851. Tot op heden is er nog geen exacte afsluitdatum gekend.

Stock		
ID	SHARETYPE	STOCKEXCHANGE
---	-----	-----
1	1	1

Stock_name			
STOCK	NAME	STARTDATE	ENDDATE
---	-----	-----	-----
1	Banque Nationale (°1851)	16-JAN-1851	31-DEC-3999

De inhoud van NAME bevat echter nog meer info dan enkel de naam. Dit is bewust gedaan, voor meer duidelijkheid. Zo vermelden we het eerste jaar dat dit aandeel op de beurs is verschenen. Gewoonlijk wordt er zelfs nog meer info meegegeven, en dit omdat een gebruiker op zijn invoerscherm enkel de “name” zal zien, en door deze truck dus nog extra informatie ter beschikking heeft. We illustreren dit aan de hand van een ander aandeel met intern nummer 211.

Stock_name			
STOCK	NAME	STARTDATE	ENDDATE
---	-----	-----	-----
211	Banque Travaux Publiques (des) (société) (°1872)	02-JAN-1872	31-DEC-3999

Elders in de databank zal er natuurlijk enkel gewerkt worden met de code voor deze aandelen ingevoerd door SCOB, namelijk 1 en 211.

Tegelijk met het toewijzen van een unieke SCOB-stockcode wordt er ook een temporeel veranderlijke noterings sleutel gegeven. Dit laat ons toe om bijvoorbeeld van sector te veranderen. Deze noterings sleutel is terug te vinden in de tabel Notation. De koersen die er later voor een bepaald aandeel ingevoerd gaan worden, zullen niet gekoppeld worden aan dat aandeel zelf, maar wel aan de bijbehorende noterings sleutel. We tonen voor de stocks 1 en 211 hun noterings sleutels.

Notation				
ID	STOCK	SECTOR	STARTDATE	ENDDATE
---	-----	-----	-----	-----
1	1	68	16-JAN-1851	01-JAN-1873
309	211	68	02-JAN-1872	12-JAN-1889

De koersen zelf worden opgenomen in de tabel `Notation_price`.

We tonen als voorbeeld enkele noteringen voor de aandelen met SCOB-codes 1 en 211. Deze horen dus bij noterings sleutels 1 en 309.

Notation_price										
NOT.	DAY	ARG.	PAP.	OPEN	CLOSE	MIN	MAX	PREV.	EXTERN	P
---	---	---	---	---	---	---	---	---	---	---
1	31-JAN-1851								1190	N
1	31-MAR-1851								1185	N
1	28-APR-1851								1175	N
1	26-MAY-1851								1170	N
1	30-JUN-1851								1170	N
1	31-JUL-1851								1185	N
309	29-JAN-1872								548	N
309	31-JAN-1872	550		547.5	547.5	547.5	550			N
309	26-FEB-1872								542.5	N
309	29-FEB-1872			542.5	544					N
309	25-MAR-1872								547.5	N
309	30-MAR-1872			575	575	570	575			N
309	29-APR-1872								575.5	N
309	30-APR-1872			575						N

Elk aandeel of elke obligatie (elk financieel product dat terug te vinden is in de tabel `stock`) heeft natuurlijk ook een bijbehorend type. Dit typenummer is temporeel veranderlijk en is terug te vinden in de tabel `Stock_type`. We tonen voor de stocks met nummer 1 en 211 hun types.

Stock_type			
STOCK	TYPE	STARTDATE	ENDDATE
-----	-----	-----	-----
1	17	01-JAN-1000	31-DEC-3999
211	10	01-JAN-1000	31-DEC-3999

Om te weten wat elk nummer (17 en 10) wil zeggen, kijken we in de tabel Stocktype.

Stocktype	
ID	NAME
17	part sociale (Maatschappelijk aandeel)
10	action de capital (kapitaalaandeel)

We merken op dat er in de databank zoveel mogelijk gewerkt wordt met de originele benamingen uit de koersboeken. Dit komt in de praktijk neer op hoofdzakelijk Franse benamingen. We hebben hier bewust voor gekozen (ook al is de voertaal van SCOB Nederlands) omdat dit de herkenbaarheid verhoogd voor de operators. Voor alle duidelijkheid wordt de Nederlandse uitdrukking er bij vermeld tussen haken.

Elk aandeel wordt nu gekoppeld aan het bedrijf dat dit aandeel heeft uitgegeven. Deze koppeling gebeurt in de tabel stock_corporation. De benaming van het bedrijf in kwestie is terug te vinden in de tabel Corporation_name.

We illustreren met stock 1 en 211. Deze horen bij de respectievelijke bedrijven 349 en 619.

Stock_corporation			
STOCK	CORPORATION	STARTDATE	ENDDATE
1	349	16-JAN-1851	31-DEC-3999
211	619	01-JAN-1000	31-DEC-3999

Corporation_name			
CORP.	NAME	STARTDATE	ENDDATE
349	Banque Nationale de Belgique (Nationale Bank van België)	05-MAY-1850	31-DEC-3999
619	Banque des travaux publics	22-APR-1871	01-JAN-1887

3. De Bedrijfsmodule

De bedrijfsmodule is een module die geschreven is om de gegevens van de verschillende bedrijven in te voeren, maar ook om op een gebruiksvriendelijke en handige manier de informatie die er in de databank zit terug op te vragen. De gevonden info kan opgeslagen en afgedrukt worden.

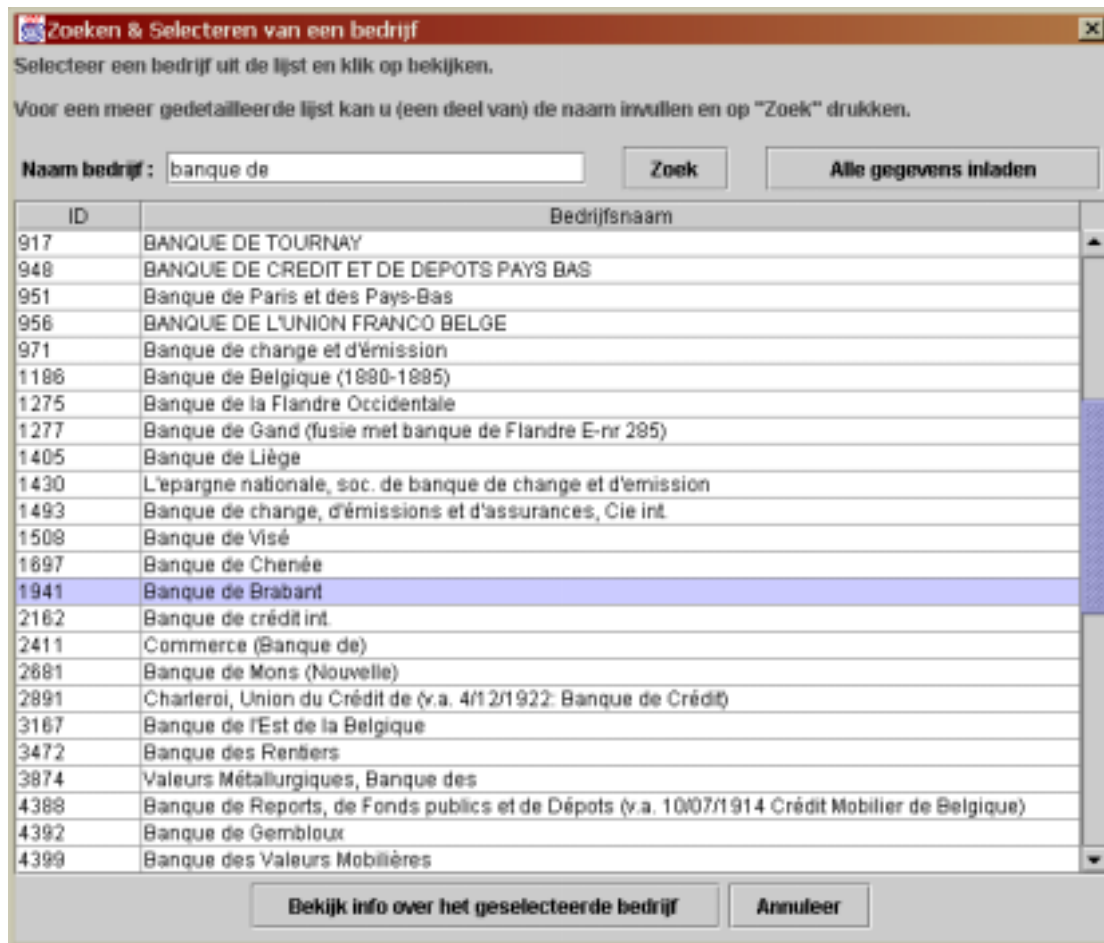
Bij het opstarten van de module verschijnt er volgend scherm.



De eerste stap die men nu doet, is het bedrijf kiezen waarvoor men gegevens wil invoeren, of waarvan men gegevens wil opvragen. Dat gebeurt met de knop `Selecteer`.

Door hier op te klikken verschijnt er een schermje waarmee je kan zoeken op de naam van een bedrijf (de eerste letters ingeven is voldoende) of waarmee je alle bedrijven die er in de databank zitten kan opvragen.

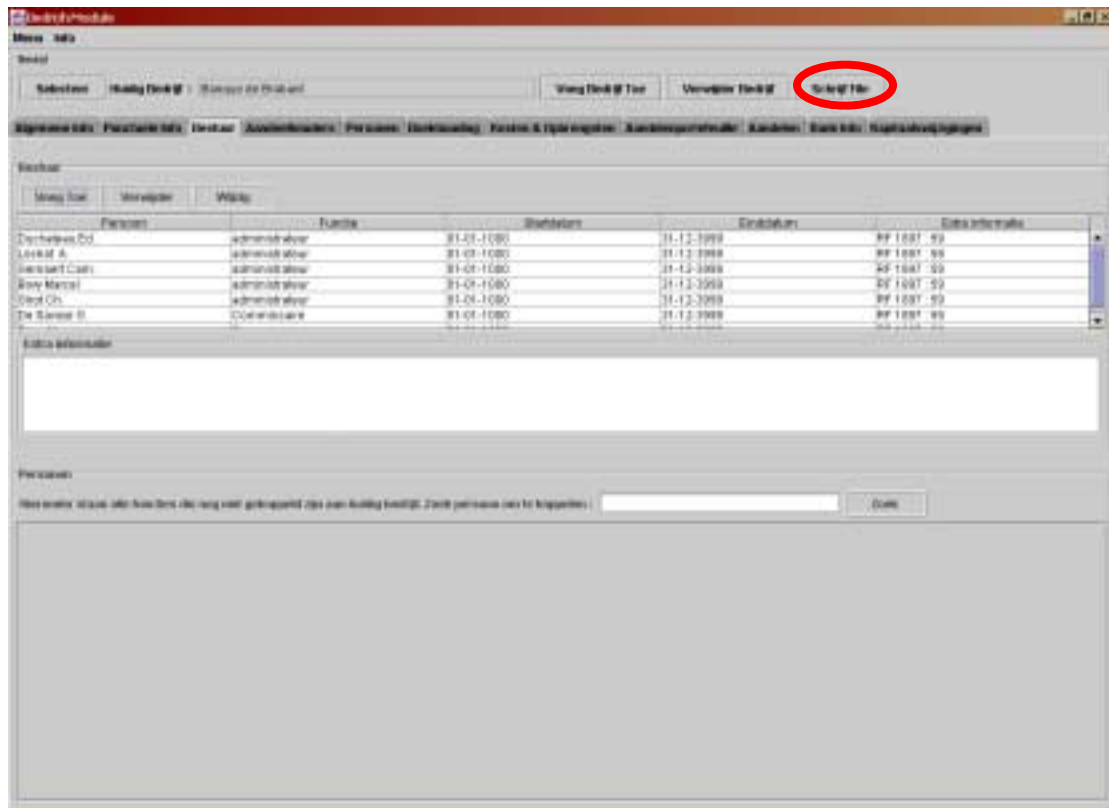
Als voorbeeld zoeken we naar `Banque de Brabant`.



Als we nu klikken op **Bekijk info over het geselecteerde bedrijf** keren we terug naar het opstartscherm, maar deze keer zijn de tabbladeren ingevuld met de gegevens van het net gekozen bedrijf.

De tabbladeren spreken voor zichzelf. Er zijn er 11. Per tabblad zijn er een aantal functies die relevant zijn voor het desbetreffende onderwerp dat er op dat tabblad behandeld wordt en die met een simpele muisklik ingevuld kunnen worden.

Als sfeerbeeld tonen we bijvoorbeeld het tabblad **Bestuur**. Hier vinden we alle info terug i.v.m. het bestuur van de "Banque de Brabant", kunnen we gegevens verwijderen of net toevoegen, met eventueel nog opmerkingen. Al deze gegevens worden op een overzichtelijke manier op het scherm voorgesteld.



Wil je de gevonden info in een handige en compacte vorm bij elkaar opslaan, dan klik je op **Schrijf File**. Je kan dan kiezen hoe je de file gaat noemen en waar je deze wil opslaan.

We geven een sfeerbeeld van het resultaat voor "Banque de Brabant".

```

Informatie betreffende bedrijf "Banque de Brabant" (ID = 1941)

Oprichting : '12-12-1889'
Afmakening : '01-01-1999'

NAAM VAN HET BEDRIJF :
-----
Naam : Banque de Brabant
SECTOR : 12-12-1889
ELEKTOR : 01-01-1999

Baanopleggingen van het bedrijf :
-----
Adres : Rue des Douze-Apôtres 26
Land : België
SECTOR : 12-12-1889
ELEKTOR : 01-01-1999

Vestigingen van het bedrijf :
-----

Administratieve acties van het bedrijf :
-----

Functionele informatie van het bedrijf :
-----
Datum : 12-12-1889
Functionele informatie : Algemeen Verslag van de Tweede Kamer van de Staten-Generaal (EF 1007:99)

```

De achterliggende "handelingen" die er in de databank plaatsvinden worden nu getoond.

Bedrijven die aangemaakt worden via "de Bedrijfsmodule" (of rechtstreeks via "Access") verschijnen in de tabel Corporation en de tabel Corporation_name. In Corporation wordt er aan elk bedrijf een unieke sleutel (ID) gegeven, de koppeling met de naam gebeurt in Corporation_name.

Corporation		
ID	STARTDATE	ENDDATE
-----	-----	-----
1941	12-DEC-1889	01-JAN-1899

Corporation_name			
CORPORATION	NAME	STARTDATE	ENDDATE
-----	-----	-----	-----
1941	Banque de Brabant	12-DEC-1889	01-JAN-1899

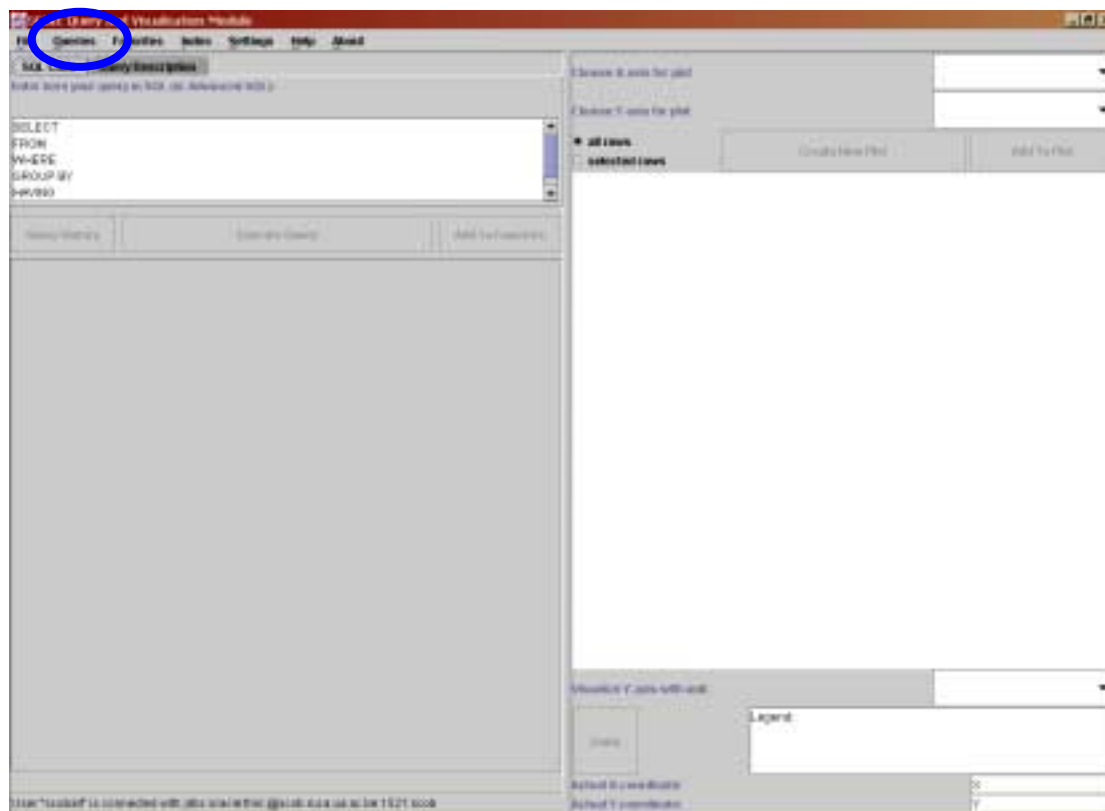
Aan de constructie van de tabellen (STARTDATE - ENDDATE) is te zien dat de bedrijfsnaam temporeel kan veranderen.

Enmaal een bedrijf is aangemaakt, kunnen we allerlei informatie hieraan koppelen, zoals aandelen en obligaties, bestuurders, boekhoudingen enz. Hiervoor worden hoofdzakelijk de Corporation_* tabellen gebruikt. Voor meer uitleg verwijzen we naar **6.2**.

4. De Outputmodule

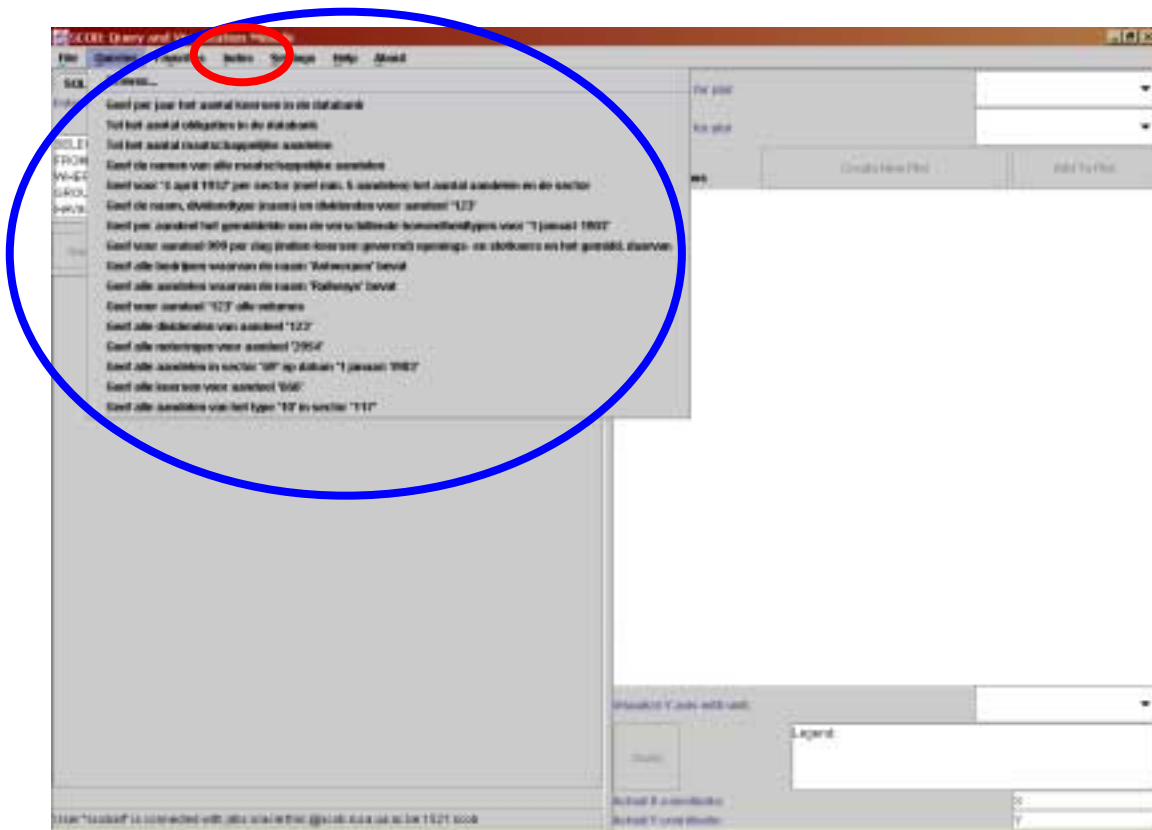
De volgende module die we bespreken is de zogenaamde "Outputmodule" of "Grafische module". Deze dient om een "beeld" (letterlijk en figuurlijk) te krijgen van de data die in de databank zitten.

Bij het opstarten van deze module verschijnt er volgend scherm.



Er is een uitgebreide statusbalk, met verschillende functionaliteiten. Naast de traditionele mogelijkheden (waaronder Help en About waaronder je een eenvoudige helpfunctie kan terugvinden en de namen van de personen die deze module ontworpen hebben en waarvoor de module dient) zijn er een aantal gespecialiseerde keuzemogelijkheden die typisch zijn voor onze databank. Zo kan men een aantal vooropgestelde queries op de databank stellen. Men klikt op **Queries** en een uitgebreide lijst mogelijkheden verschijnt. Hieruit kiest men dan de gewenste query en het resultaat wordt door de module gegenereerd.

We geven een sfeerbeeld van de keuzelijst queries.



Een andere, zeer belangrijke mogelijkheid, is de **berekening van indexen**, (returnindex of koersindex). Dit kan men terugvinden onder **Index**. De gebruiker doorloopt dan een wizard waarin hij belangrijke keuzes moet maken i.v.m. de indexberekening, zoals of het bijvoorbeeld over aandelen of obligaties gaat, en welke precies uit welke sectoren en van welke bedrijven (Belgische of buitenlandse of combinaties ...). Er moet ook aangegeven worden of men van de termijnmarkt berekeningen wil maken of van de contantmarkt, of dat op maandelijkse of op jaarlijkse basis moet gebeuren en of de index "gelijkgewogen" moet zijn of met "marktkapitalisatie" en al dan niet met of zonder dividenden. Van deze werkt de maandelijkse index perfect, de jaarlijkse index moet nog aangepast worden. Tenslotte moet men de tijdsperiode ingeven waarover de berekening moet lopen. Men krijgt ook de mogelijkheid om de logfile op te slaan, en daarin kan men dus alle gegevens en tussenstappen van de berekening terugvinden. Deze logfile is de ideale hulpbron om achteraf de berekening in stukken nog eens te bekijken en eventuele "rariteiten" op te sporen en te kunnen verklaren.

Het hoofdscherm van de Outputmodule bestaat uit twee helften, waarvan de **linkerhelft** bestaat uit **twee tabbladeren** SQL en Query Description. Zij betekenen hetzelfde, alleen geeft de SQL-kant SQL de "echte" vorm van de query terwijl Query Description de beschrijving van dezelfde query in "mensentaal" is. De query wordt dan uitgevoerd door op **Execute Query** te klikken.

De **rechterhelft** van het scherm is het eigenlijke "outputscreen" waarop de grafieken te voorschijn gaan komen als resultaat van de gestelde query.

Als voorbeeld berekenen we de volgende query:

```
SELECT day, price
FROM scobresearch.stock_info
WHERE stock = 607
```



Dit geeft een grafiekje terug voor de stock met nummer 607, Charbonnage d'Hornu et Wasmes (actions de dividende), waarvan we hier een uitvergroot gedeelte tonen.

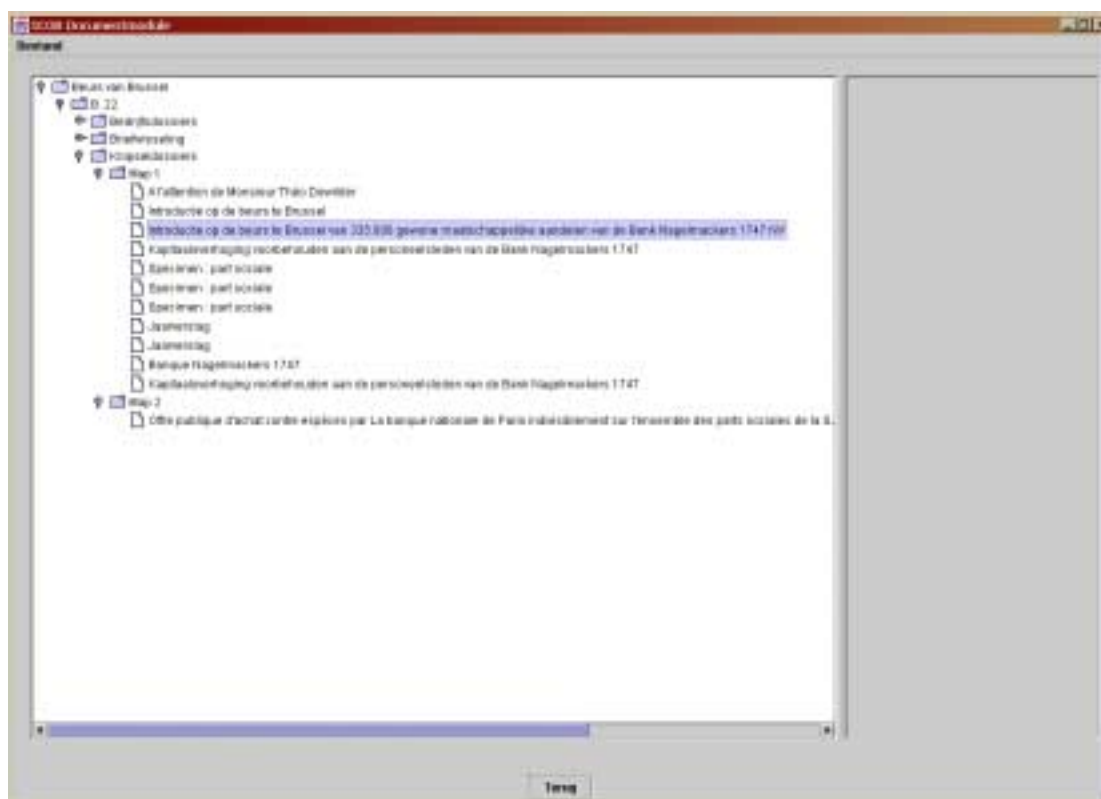
Het resultaat van een query kan natuurlijk bewaard worden, hetzij als tekstbestand, hetzij als Excel-bestand (enkel terug te openen in Excel 2000).

De berekeningen die in deze module worden doorgevoerd steunen hoofdzakelijk op de tabel `stock_info`. Hierop wordt dieper ingegaan in 6.

5. De Documentmodule

De documentmodule is een module die geschreven is voor het **inventariseren** van onze uitgebreide inventaris van gegevens. De module maakt het mogelijk om op een eenvoudige manier nieuwe archieven in de databank in te voeren en ook om gegevens op te zoeken in de reeds geklasseerde archieven.

Een voorbeeld van de nu reeds bestaande (doch niet meer gebruikte) documentmodule:



Op dit moment wordt de Documentmodule volledig herschreven. Vertrekkende van de reeds bestaande interface zal de module totaal herwerkt worden tot een nieuw en functionerend geheel.

Het wordt dus een volledig nieuwe module, geschreven in JAVA en geïnspireerd door de reeds bestaande module, waarmee alle documenten van het SCOB (knipsels, prospectussen, koersboeken, kranten,...) op een overzichtelijke wijze geklasseerd en opgevraagd kunnen worden. De module moet eenvoudig op alle SCOB-computers geïnstalleerd kunnen worden en moet gebruikt kunnen worden door mensen die geen enkele ervaring hebben met dit soort programma's. De referenties naar gegevens, die nu verzameld zijn in grote dozen, kunnen dan op een overzichtelijke wijze in de databank ingevoerd worden door jobstudenten en natuurlijk ook weer opgevraagd worden door onderzoekers, die hiermee op een eenvoudige wijze de desbetreffende doos met knipsels kunnen terugvinden.

6. MakeStockInfo

Dit is een module die geen echte interface op het scherm tovert, maar eentje die wel een zeer intensieve en krachtige berekening start "in de achtergrond" en die alle beschikbare info verzamelt over de stocks in de databank aanwezig.

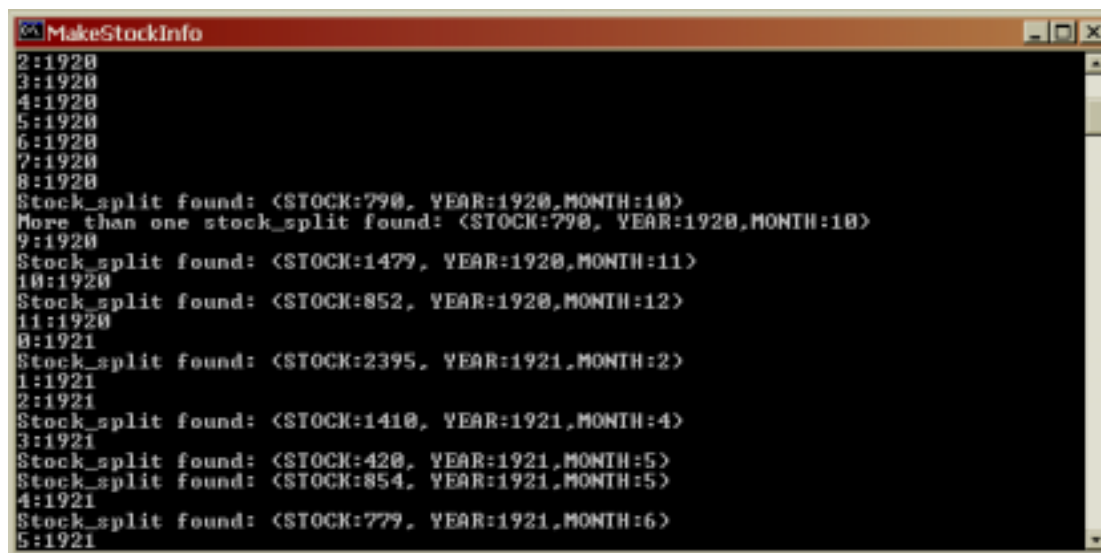
De module MakeStockInfo genereert de tabel `Stock_info` en de bedoeling hiervan is in feite **drievoudig**. Per aandeel zijn er op een dag meerdere koersen ter beschikking. MakeStockInfo kiest hier steeds de openingskoers uit en als deze niet voor handen is wordt er d.m.v. een zekere interpolatie een andere beschikbare koers gemaakt. De eerste toepassing van de module is dus per aandeel op een dag **één koers** kiezen die verder gebruikt gaat kunnen worden in bijvoorbeeld de indexberekeningen. Net zoals er voor een aandeel op een dag meerdere koersen kunnen zijn, kunnen er ook per aandeel op een dag meerdere volumes of hoeveelheden in de databank terug te vinden zijn. Ook hier kiest de module er eentje uit, wat dus de tweede bedoeling van de module is: **één volume** ter beschikking stellen van de gebruiker. De derde bedoeling is **alle info over stock splits verzamelen** uit de tabel `Stock_splits` en dit op een overzichtelijke manier representeren in de tabel `Stock_info`. Op die manier groepeert de tabel `Stock_info` alle nuttige en beschikbare informatie over alle stocks in een enkele tabel.

Bij het opstarten verschijnt er een klein schermje waarin de gewenste periode afgebakend kan worden waarover de berekening zou moeten lopen.



Met een muisklik op `start` begint de berekening te lopen. Deze kan eenvoudigweg naar de achtergrond geklikt worden, en kan alzo rustig rekenen terwijl de pc-gebruiker zijn andere taken op de computer tot een goed einde kan brengen.

Als controle dat de module effectief nog berekeningen aan het uitvoeren is, wordt er in een venstertje de datum die op dat moment behandeld wordt afgedrukt plus de eventuele stock splits die er zijn opgetreden. De exacte en uitgebreide informatie is dan terug te vinden in de tabel `stock_info` zelf, de "echte" output van de module.



De output van deze berekening is een tabel `stock_info` waarin alle info staat over de stocks, met stock splits en hun bijbehorende gegevens zoals bijvoorbeeld de datum. Deze tabel `stock_info` bestaat uit een selectie van data uit andere tabellen uit de databank.

We geven hier een sferbeeld:

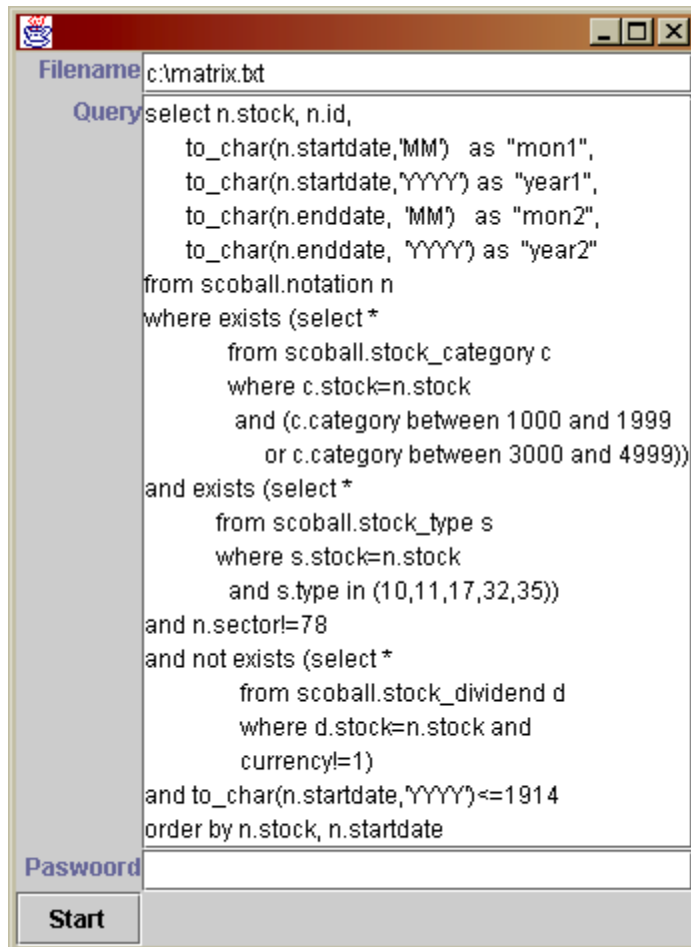
Stock_info											
STOCK	MONTH	YEAR	PREV_PRICE	PREV_QUANT	PREV_MEICH	PRICE	DEVIDEND	DAY	PRICE_RETURN	EQVOLUME_D	INVERSE_PR
9	Z	1920	345	4000	.000140430	345	0	01-FEB-1920	1	1	.011050201
10	Z	1920	438	9400	.000140870	438	0	01-FEB-1920	1	1	.012298881
15	Z	1920	697.5	10000	.000272904	697.5	0	01-FEB-1920	1	1	.011430692
25	Z	1920	1900	112500	.008393824	1937.5	0	01-FEB-1920	1.01973684	1.01973684	.030506316
28	Z	1920	8150	7500	.001516779	4405	0	01-FEB-1920	.852328006	.852328006	.030194175
29	Z	1920	748.75	9000	.000244626	640	0	01-FEB-1920	.85475759	.85475759	.011215559
31	Z	1920	7850	4000	.001107396	7300	0	01-FEB-1920	1.03546099	1.03546099	.030141844
33	Z	1920	7780	4200	.001276217	7700	0	01-FEB-1920	.992846287	.992846287	.030129222
36	Z	1920	8600	5900	.001699592	8100	0	01-FEB-1920	.941800465	.941800465	.030116279
48	Z	1920	4800	18000	.004894814	6130	0	01-FEB-1920	.946133946	.946133946	.030133946
49	Z	1920	1705	4000	.000247617	1875	0	01-FEB-1920	1.09970474	1.09970474	.03019451
74	Z	1920	8100	31000	.008660835	7750	0	01-FEB-1920	.956790123	.956790123	.030123457

Ook deze module wordt op dit ogenblik herschreven, gezien de hogere eisen die de economen aan de outputdata stellen.

7. MakeMatrix

Ook dit is een module die geen spetterende interface heeft, maar het wel moet hebben van zijn rekenkracht. Deze module berekent, zoals de naam reeds doet vermoeden, een matrix, die gebruikt kan worden voor verder onderzoek door economen i.v.m. portefeuilles.

Het opstartscherm van de module:



Deze module is speciaal ontwikkeld voor verder onderzoek door economen met de bedoeling te kunnen nagaan in welke mate het rendabel is om portefeuilles samen te stellen op basis van fundamentele karakteristieken van beursgenoteerde ondernemingen. "MakeMatrix" genereert hiertoe een gegevensbestand (*matrix.txt* in het voorbeeld) dat de onderzoeker de kans geeft bovenstaande analyses uit te voeren.

Het gegenereerde databestand groepeert per onderneming een reeks originele, ononderbroken maandobservaties van relevante kenmerken,

zoals bijvoorbeeld koersen, dividenden en aantallen. Deze komen (via de tabel `stock_info`), rechtstreeks uit de SCOB-databank. Op basis van deze gegevens worden dan afgeleide variabelen berekend zoals return, dividendrendement en ondernemingsgrootte.

Op basis van deze fundamentele karakteristieken worden de bedrijven dan gerangschikt en de bijbehorende portefeuilles samengesteld. Om het jaar worden deze herschikt om het portefeuillekenmerk te behouden zodat bijvoorbeeld jaar na jaar de kleinste ondernemingen effectief in de portefeuille met de kleinste aandelen zitten. Op deze manier kunnen dan lange-termijnreturns tussen verschillende portefeuilles vergeleken worden en conclusies getrokken worden.

8. De Logboeken

Er zijn een aantal logboeken aangemaakt in Word, die bedoeld zijn als **documentatie**. Er is geen onmiddellijke, directe koppeling wat betreft de inhoud tussen de logboeken en de databank, dus het is best mogelijk dat er onderlinge tegenstrijdigheden kunnen voorkomen. De enige band die er bestaat tussen de logboeken en de databank is dat de data uit de databank **manueel** in het logboek op een gebruiksvriendelijke en overzichtelijke manier is ondergebracht.

8.1. Logboek Aandelen

Dankzij het "Logboek Aandelen" kunnen we een **globaal overzicht** behouden op de databank en op alle hierin genoteerde aandelen.

In dit logboek is slechts een minimaal gedeelte van de informatie uit de databank overgebracht. Het gaat hier enkel over bedrijfsinformatie en aandeleninformatie.

Van de **bedrijven** worden start- en einddatum weergegeven, net als de sector waarin ze werkzaam zijn en het interne SCOB-codenummer (uit `Corporation` en `Corporation_name`). Op deze manier wordt de historie van deze bedrijven beschreven waarbij er in het bijzonder de fusies bijgehouden worden en de liquidaties. Bedrijven worden bij elkaar geplaatst indien ze in de loop der tijd gefusioneerd worden. Zo krijgt de

gebruiker van het logboek dus een chronologisch historisch overzicht van de bedrijfsevolutie.

Van de **aandelen** wordt er eerst het SCOB-codenummer (uit Stock en Stock_name) genoteerd, het typenummer (uit Stock_type en Stocktype) en de start- en einddatum. Er wordt aangegeven of ze op de termijnmarkt (TM) genoteerd stonden en of het over een eventueel tijdelijk genoteerd aandeel (TG) gaat.

We geven een sfeerbeeld uit het "Logboek Aandelen".

BEDRIJVEN				GENOTEERDE AANDELEN			
NAAM BEDRIJF START- EN EINDDATUM SECTOR + CODE LAND	CODENAAM	BEDRIJVENAAM	TYPE OPERATIE OF BEDRIJVEN	CODE + TYPE AANDEEL START- EN EINDDATUM NOTERING	INFORMATIE OVER DIT AANDEEL	OMLAANDELEN DE BLOK Per pagina	Capitaalstructuur & Titre Multiple & aandelen Termijnmarkt (TM) die op beide markten noteren
Aachener-Hütten-Actien-Verein							
Aachener Hütten-Actien- Verein 22/10/1868 - Beeld. 1985			Eerste Merken reeds in 1847	690 act. de capital 5/12/1922 - 30/3/1926		EF 1886 1487	
Abattoirs et Marchés d'Anderslecht							
Abattoirs et Marchés d'Anderslecht 1/1/1885 - 18/5/1922 Wording: 1896			1922 Liquidated	1400 act. per lignes 28/6/1904 - 3/04/1922 1400 act. estimes 28/6/1904 - 3/04/1922		Keering 1927 5	
Aby							
Manufacture et Construction Heavy General Contractors			1978 Opleiding van IED			1886 1922	
ETE Electrification et Thermaic Systems							

We zien dus dat in het "Logboek Aandelen" de genoteerde aandelen aan het uitgevende bedrijf gekoppeld worden. Er zijn immers twee blokken (BEDRIJVEN en GENOTEERDE AANDELEN) die elk nog verder onderverdeeld worden in 4 subblokken.

In het blok GENOTEERDE AANDELEN geeft de eerste kolom CODE + TYPE AANDEEL de startdatum, de einddatum en de notering van het desbetreffende **aandeel**. Deze data zijn altijd correct omdat ze rechtstreeks uit de databank komen.

De start- en einddatum van de **bedrijven** (in BEDRIJVEN, de eerste kolom) zijn echter niet altijd gekend. Het jaar is gewoonlijk wel gekend, en dan neemt men standaard "1/1/gekende startjaar" als startdatum (deze datum geeft duidelijk aan dat de startdatum niet exact gekend is

omdat er op 1 januari immers geen bedrijven opgericht worden) en "31/12/gekende afsluitjaar" als einddatum (ook deze datum spreekt voor zich want op 31 december worden er geen bedrijven afgesloten). Indien zelfs de jaren niet gekend zouden zijn, wordt er standaard gewerkt met "1/1/1000" als startdatum en de einddatum wordt gewoon opengelaten om aan te geven dat deze niet gekend is (zie als voorbeeld Aachener-Hütten-Actien-Verein, waar de exacte einddatum ontbreekt).

8.2. Logboek Obligaties

Net zoals het "Logboek Aandelen" bestaat er een "Logboek Obligaties", dat steunt op dezelfde opvattingen maar is toegespitst op obligaties. Ook hier is de hoofdbedoeling een gebruiksvriendelijk en overzichtelijk document te creëren waarin in één oogopslag alle belangrijke informatie wat betreft obligaties teruggevonden kan worden.

BELGIE : Caisse d'amortés dues par l'état 4 1/2 %							
Caisse d'amortés dues par l'état 16/11/70 - Banque: 615			Niet edit een bedrag/ Opgevoerd in 1870 (aankoop speculatiehandel)	13038 Caisse d'amortés dues par l'état (4 1/2 %) 2046871 - 2201892 wordt voortgezet door 13038 Caisse d'amortés dues par l'état serie A (4 1/2 %) 2206892 - 21.1.191 en 13392 Caisse d'amortés dues par l'état serie B (4 1/2 %) 2206892 - 30/1/192 Terug overgenomen op 3/2/192	Annuit Juli Vanaf 29/5/1903 wordt dit Serie B	BF 1903-3	
				13075 Belgique: Amortés dues par l'état (4 1/2 %) (serie A) (aankoop met bezwaar en januari 1957) 2956923 - 12/03/1966	Overdracht van obligatie 13039		
BELGIE : Caisse d'amortés dues par l'état 4 %							
				13042 Caisse d'amortés dues par l'état (4 %) 2056873 -	Vanaf 29/5/1903 wordt dit Serie A		
				13077 Belgique: Amortés dues par l'état (4 %) (serie A)	Overdracht van obligatie 13042		

8.3. Logboek Stock Splits

In dit logboek wordt alle informatie opgenomen i.v.m. stock splits, reverse splits, bonusaandelen, ruil van aandelen bij fusies enz. Er wordt verwezen naar de bron van de split, naar de exacte ruilverhoudingen en de datum van de verwerking. Er wordt ook aangegeven of het verwerkt is of niet verwerkbaar is. Er wordt ook een kleine tabel afgedrukt met koersen, indien de split verwerkt is. Deze koersen komen m.b.v. SQL rechtstreeks uit de databank, uit de tabel `Notation_price`.

Een sferbeeld:

NAAM AANDEEL	EVOLUTIE VAN KOERS, AANTAL AANDELEN, NOMINALE WAARDE + BRON		AAN DEEL NUMMER	FACTOR van de STOCK/REVERSE SPLIT of RUILVERHOUDING	DATUM
3220 : Sofobel					
STOCK	ID	STARTDATE	ENDDATE		
3219	3966	29-10-28	05-11-28		
3219	3968	05-11-28	23-03-31		
3219	13965	23-03-31	31-05-31		
STOCK	ID	STARTDATE	ENDDATE		
3220	3967	29-10-28	05-11-28		
3220	3969	05-11-28	21-06-37		
Sofobel	UIT access stock_quantity		3220 art ord	Reverse Split : 15 : 2	2903/1931
	In 1931 verheijgt aandeel 3219 van de beurs wegens eenmaking tot part sociale			60.000 pnb => 60.000 p. soc 60.000 ord => 8.000 p. soc. 30.000 p. bond => 9.000 p. soc. 15.000 dty => 3.000 p. soc. (aandeel over 34.000 in RF)	VERWERKT OP 2303/1931
3969	31-10-30			24	24,5
3969	28-11-30			17	17
3969	31-12-30			15	
3969	30-01-31			11	12
3969	27-02-31				20
3969	23-03-31			100	
3969	31-03-31			119	116
3969	30-04-31			102	102
3969	29-05-31			100	100
3969	30-06-31			74	78
3969	31-07-31			82	82
3969	31-08-31				78
					60
STOCK	SPLITDATE	BEFORE	AFTER	SOURCE	TOSTOCK
3220	2303/31	15	2	Reverse Split Sofobel. Eenmaking aandelen tot part sociale	3220