MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN

INTERNATIONAL INFORMATION TECHNOLOGY UNIVERSITY

COMPUTER ENGINEERING AND INFORMATION SECURITY DEPARTMENT

COURSE WORK

Designing Databases: Introduction to SQL

Done by student Kubenov Timur

Group CSSE-1703K

Almaty 2018

**An Overview of Sapienza University Library Case Study**

Sapienza University Library is an academic library that holds more than 200,000 items that could only be handed out to students for a limited time, to a maximum of 3 months, excluding textbooks. The library database is not allowed to share the university system database. The staff is assigned and chosen by the university management, although the library is located in a separate building.

The process of leasing an item is considered rather simple: from a wide range of items, the student is allowed to choose any item they wish, up to 5 items at a time. The items, and the student’s ID card are then scanned by a library staff member, assigning the items to the student in the database. The leasing time is set by default depending on the type of the item, but the students are able to change it, as long as it doesn’t exceed the default time.

The students are also allowed to extend the lending time, but only on the condition of the library staff member allowing it. The extra lending time cannot exceed the half of the maximum lend time.

In case of a student returning the item late or not returning it at all, a fine will be charged for every week past the return date. The weekly penalty charge is 3 euros, which must be paid to the university management staff, unless the student has a strong reason not to return the item before the time due.

If the student claims to have lost the book, they have to pay twice the price of the item. In the case claim was also belated, the student has to pay weekly charges for the duration of time it was late.

Students that have been acting inappropriately or haven’t paid their dues for a time period exceeding 3 months will be put in a blacklist. Blacklisted students won’t be able to take items from the library until they have paid their fines.

The database should contain full information on the items, while the information on the students is stored in the university database. The registration of a lend requires only the IDs of the student and the item and the lend period, the rest of the information should automatically show up in the registration form.

To register an item its type, genre, author name, name, publishing company name, and the publishing date are required.

Data Requirements and Constrains

*Staff*

The data required on members of staff includes staff number, name (first and last name), position, sex, date of birth (DOB).

*Items*

The data required to register an item require information on item’s ISBN, title, author, publishing date, publishing house, position in the library listed as bookshelf and rack numbers, and the state of an item, which is used to define the item’s availability

*Clients*

There are two main types of clients that register in the Sapienza University Library: students and teachers. To register in the library database, they have to give their full name and phone number. Clients can be blacklisted for their inappropriate behavior on the premises of the library. Clients belating the return of books will be fined. Each client is given a unique ID number.

*Rents*

Client that have rent a book withdraw a rent that includes the item data, client data and dates of rent initiation, rent expiry date and rent end date.

Transaction Requirements (Sample)

Book data:

* ISBN
* Author name
* Book name
* Publishing company
* Publishing date
* Status

Client data:

* id
* Full name
* Phone number

When item is taken by clients, change the inform of items.

Book state includes:

* Book information
* Book status
* Book amount

Data query:

* Book ID
* Book name
* List of people who took the book
* Students’ blacklist
* Students’ IDs
* Students’ full names
* Books taken by the student
* Teachers’ IDs
* Teachers’ full names
* Teachers’ addresses
* Books taken by the teacher

2 BUILDING A CONCEPTUAL DATA MODEL FOR THE *DREAMHOME* CASE STUDY

*Step 1. Identify entities*

Table 1. Entities of the conceptual data model

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity name** | **Description** | **Aliases** | **Occurrence** |
| Staff | General term describing all staff employed by *Sapienza University Library* | Employee | Each member of staff works at the library |
| Item | General term describing all items for lending | Book | Every single item is available for taking on the premises of the library. Staff manages all of the Items. |
| Client | General term describing all clients of the *Sapienza University Library* | Customer | Each client is registered in the library database. Any client, including teachers and students, can lend multiple Items. |
| Rent | General term describing all processes of renting a Item by clients | Lease | Every time a client rent an item a rent is drawn up, any client can take desired amount of Items, but they may not be shared. |

Table 2. Relationships of the conceptual data model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Multiplicity** | **Relationship** | **Multiplicity** | **Entity Name** |
| Staff | 1..1 | DrawsUp | 0..\* | Rent |
| Client | 1..1 | Holds | 1..\* | Rent |
| Item | 1..1 | AssociatedWith | 0..\* | Rent |

***Library entities***

**Staff** staffNo, name (composite: fName, lName), position, sex, DOB

**Item**  ISBN, author, name, publishingDate, publishingHouse,

libPosition (composite:bookshelfNo, rackNo), state

**Client** clientNo, name (composite: fName, lName), telNo, blacklisted,

debtValue, class

**Rent** rentNo, rentStart, rentEnd, rentOver, isLate, duration

(derived as rentOver – rentStart)

Table 3. Attributes of the conceptual data model

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Data Type, Length** | **Nulls** | **Multi-valued** | **Composite** | **Derived** | **Default** |
| Staff | staffNo  name  fName  lName  position  sex  DOB | Uniquely identifies a member of staff  Name of staff  First name of staff  Last name of staff  Job title of member of staff  Gender of member of staff  Date of birth of member of staff | 5 characters  15 characters  15 characters  10 characters  1 character (M or F)  Date | No  No  No  No  Yes  Yes | No  No  No  No  No  No | No  Yes  No  No  No  No  No | No  No  No  No  No  No | No  No  No  Assistant  No  No |
| Item | ISBN  author  item\_name  publishingDate  publishingHouse  libPosition  bookshelfNo  rackNo  state | Unique book number  Author of the book  Name of the book  Publishing date of the book  Publishing house of the book  Position in the library  Number of the bookshelf  Number of the rack  Denotes the state of the book | 13 characters  20 characters  30 characters  Date  15 characters  3 digits integer  4 digits integer  10 characters | No  Yes  No  Yes  Yes  No  No  No | No  Yes  No  No  No  No  No  No | No  No  No  No  No  Yes  No  No  No | No  No  No  No  No  No  No  No | No  No  No  No  No  No  No  Available |
| Client | clientNo  name  fName  lName  telNo  blacklisted  debtValue  class | Uniquely identifies a client  Name of client  First name of client  Last name of client  Telephone of client  Position in the blacklist  Value of the debt  Shows position of the client | 5 characters  15 characters  15 characters  12 characters  1 character (Y or N)  4 digits integer  10 characters | No  No  No  No  No  Yes  No | No  No  No  Yes  No  No  No | No  Yes  No  No  No  No  No  No | No  No  No  No  No  No  No | No  No  No  No  N  Null  No |
| Rent | rentNo  rentStart  rentEnd  rentOver  isLate | Uniquely identifies a particular lease  Start rent date  Finish rent date  Actual rent finish date  Is the rent due time has passed | 5 characters  Date  Date  Date  1 character (Y or N) | No  No  No  No  No | No  No  No  No  No | No  No  No  No  No | No  No  No  No  No | No  No  No  No  N |

***Attributes’ domains***

The domains are determined for the attributes where the specific rules are applied.

The attribute domain of valid staff numbers (staffNo) as being a five-character variable-length string, with the first two characters as letters and the next one to three characters as digits in the range 1–999. The same domain can be applied to the client numbers and rent numbers.

The attribute domain of a number identifying a specific book is a 13-digit long integer (ISBN).

The attribute domain of a book, identifying a book’s state (state) is a 10-character long string that shows whether the book is available (Available), taken (Taken), or lost (Lost).

The attribute domain specifying the position of the client (class) is a 10-character long string that shows whether the client is a student (Student), a Teacher (Teacher) or a member of the staff (Staff).

Table 4. Key attributes of the conceptual data model

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Type of key** | **Description** |
| Staff | staffNo | Primary key | Staff number uniquely identifies a member of staff within a branch |
| Book | ISBN | Primary key | Property number uniquely identifies an item for rent |
| Client | clientNo  telNo | Primary key  Alternate key | Client number uniquely identifies a client  Telephone number may be used to identify a client |
| Rent | rentNo | Primary key | Lease number uniquely identifies a rent |

***Step 6. Validate conceptual model against user transactions***

The list of user transactions defined in the user requirements is as follows:

1. List details of all staff members

List all of the staff members that work in the library. The details are held in Staff entity which can be used to produce the required list.

1. List the details of items available for rent.

The details of items are held in Item entity which can be used to produce the required list.

1. List the books taken by a particular client.

The details of all books including their state and return date are held in the Item entity and an information about clients are held in the Client entity. The relationship Client Holds Rent can be used to produce the required list.

1. List all the books that were never returned in the last year.

The details of all books including their state are held in the Item entity which can be used to produce the required list.

1. List all the readers who have not returned a book/books for more than a week in the last month.

The details of all books including their state and return date are held in the Item entity and an information about clients are held in the Client entity. The relationship Client Holds Rent can be used to produce the required list.

1. List the clients registered.

The details of all clients registered are held in the Client entity which can be used to produce the required list.

1. List all Students/Teachers/Staff registered.

The details of all clients registered are held in the Client entity which can be used to produce the required list.

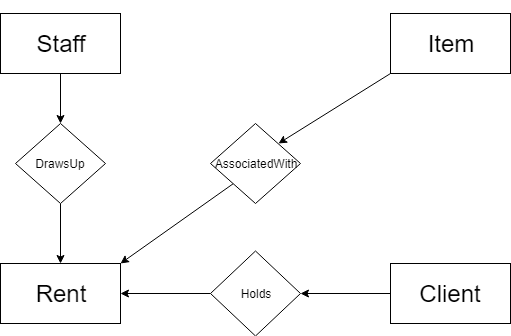


Figure 1 – ER diagram

3 BUILDING A LOGICAL DATA MODEL FOR THE DREAMHOME CASE STUDY

***Step 1. Derive relations***

**1.1 Derive relations for entities**

In the conceptual data model we identified 6 entities. For each of them we can derive relations with primary keys underlined.

**Staff (**staffNo, fName, lName, position, sex, DOB)

**Item (**ISBN, author, item\_name, publishingDate, publishingHouse, bookshelfNo, rackNo, state)

**Client (**clientNo, fName, lName, telNo, blacklisted, debtValue, class)

**Rent (**rentNo, rentStart, rentEnd, rentOver, isLate, duration)

**1.2 Derive relations for relationships**

*Staff DrawsUp Rent relationship (1:\*).*

Staff is a parent relation and Rent is a child relation. Therefore, we need to copy the primary key of the Staff relation (staffNo) to the Rent relation to act as a foreign key.

*Client Holds Rent relationship (1:\*).*

Client is a parent relation and Rent is a child relation. Therefore, we need to copy the primary key of the Client relation (clientNo) to the Rent relation to act as a foreign key.

*Item AssociatedWith* Rent *relationship (1:\*).*

Item is a parent relation and Rent is a child relation. Therefore, we need to copy the primary key of the Item relation (itemNo) to the Lease relation to act as a foreign key.

**1.3 Derive relations for multi-valued attributes**

In the Item relation the author attribute is multi-valued. Therefore, we need to create another relation ItemAuthor with author attribute and copy the primary key of the Item relation.

In the Client relation the telNo attribute is multi-valued. Therefore, we need to create another relation ClientTel with telNo attribute and copy the primary key of the Client relation.

The final list of relations with primary, alternate and foreign keys is presented in the table 1.

Table 1. Relations derived for the Sapienza University Library case study

|  |  |
| --- | --- |
| 1 | **Staff** (staffNo, fName, lName, position, sex, DOB)  **Primary key** staffNo |
| 2 | **Item (**ISBN, author, item\_name, publishingDate, publishingHouse, bookshelfNo, rackNo, state)  **Primary key** itemNo (composite: ISBN, copyNo) |
| 3 | **ItemAuthor** (author, itemNo)  **Primary key** author  **Foreign key** itemNo references Item (itemNo) |
| 4 | **Client (**clientNo, fName, lName, telNo, blacklisted, debtValue, class)  **Primary key** clientNo |
| 5 | **ClientTel** (tel, clientNo)  **Primary key** tel  **Foreign key** clientNo references Client (clientNo) |
| 6 | **Rent (**rentNo, rentStart, rentEnd, rentOver, isLate, duration, staffNo, clientNo, itemNo)  **Primary key** rentNo  **Foreign key** staffNo references Staff (staffNo)  **Foreign key** clientNo references Client (clientNo)  **Foreign key** itemNo references Item (itemNo) |

***Step 2. Validate relations using normalization***

**Staff** (staffNo, fName, lName, position, sex, DOB)

The **Staff** relation is in the 1NF, since in this relation the intersection of each row and column contains one and only one value.

The **Staff** relation is in the 2NF, since it is in the 1NF, the primary key is not composite and its every non-primary-key attribute is fully functionally dependent on the primary key.

The **Staff** relation is in the 3NF, since it is in the 2NF and there is no non-primary-key attribute which is transitively dependent on the primary key.

**Item (**itemNo (composite: ISBN, copyNo), author, name, date, publishingHouse, bookshelfNo, rackNo, state)

The **Item** relation is in the 1NF, since in this relation the intersection of each row and column contains one and only one value.

The **Item** relation is in the 2NF, since it is in the 1NF, the primary key is not composite and its every non-primary-key attribute is fully functionally dependent on the primary key.

The **Item** relation is in the 3NF, since it is in the 2NF and there is no non-primary-key attribute which is transitively dependent on the primary key.

**ItemAuthor** (author, itemNo)

The **ItemAuthor** relation is in the 1NF, since in this relation the intersection of each row and column contains one and only one value.

The **ItemAuthor** relation is in the 2NF, since it is in the 1NF, the primary key is not composite and its every non-primary-key attribute is fully functionally dependent on the primary key.

The **ItemAuthor** relation is in the 3NF, since it is in the 2NF and there is no non-primary-key attribute which is transitively dependent on the primary key.

**Client (**clientNo, fName, lName, telNo, blacklisted, debtValue, class)

The **Client** relation is in the 1NF, since in this relation the intersection of each row and column contains one and only one value.

The **Client** relation is in the 2NF, since it is in the 1NF, the primary key is not composite and its every non-primary-key attribute is fully functionally dependent on the primary key.

The **Client** relation is in the 3NF, since it is in the 2NF and there is no non-primary-key attribute which is transitively dependent on the primary key.

**ClientTel** (tel, clientNo)

The **ClientTel** relation is in the 1NF, since in this relation the intersection of each row and column contains one and only one value.

The **ClientTel** relation is in the 2NF, since it is in the 1NF, the primary key is not composite and its every non-primary-key attribute is fully functionally dependent on the primary key.

The **ClientTel** relation is in the 3NF, since it is in the 2NF and there is no non-primary-key attribute which is transitively dependent on the primary key.

**Rent (**rentNo, rentStart, rentEnd, rentOver, isLate, duration, staffNo, clientNo, itemNo)

The **Rent** relation is in the 1NF, since in this relation the intersection of each row and column contains one and only one value.

The **Rent** relation is in the 2NF, since it is in the 1NF, the primary key is not composite and its every non-primary-key attribute is fully functionally dependent on the primary key.

The **Rent** relation is in the 3NF, since it is in the 2NF and there is no non-primary-key attribute which is transitively dependent on the primary key.

***Step 3. Validate relations against user transactions***

The list of user transactions defined in the user requirements is as follows:

1. List details of all staff members

List all of the staff members that work in the library. The details are held in Staff entity which can be used to produce the required list.

1. List the details of items available for rent.

The details of items are held in Item entity which can be used to produce the required list.

1. List the books taken by a particular client.

The details of all books including their state and return date are held in the Item entity and an information about clients are held in the Client entity. The relationship Client Holds Rent can be used to produce the required list.

1. List all the books that were never returned in the last year.

The details of all books including their state are held in the Item entity which can be used to produce the required list.

1. List all the readers who have not returned a book/books for more than a week in the last month.

The details of all books including their state and return date are held in the Item entity and an information about clients are held in the Client entity. The relationship Client Holds Rent can be used to produce the required list.

1. List the clients registered.

The details of all clients registered are held in the Client entity which can be used to produce the required list.

1. List all Students/Teachers/Staff registered.

The details of all clients registered are held in the Client entity which can be used to produce the required list.

***Step 4. Check integrity constraints***

Table 2. Relations with referential integrity constraints

|  |  |
| --- | --- |
| 1 | **Staff** (staffNo, fName, lName, position, sex, DOB)  **Primary key** staffNo |
| 2 | **Item (**ISBN, author, item\_name, publishingDate, publishingHouse, bookshelfNo, rackNo, state)  **Primary key** itemNo (composite: ISBN, copyNo) |
| 3 | **ItemAuthor** (author, itemNo)  **Primary key** author  **Foreign key** itemNo references Item (itemNo) ON UPDATE CASCADE ON DELETE CASCADE |
| 4 | **Client (**clientNo, fName, lName, telNo, blacklisted, debtValue, class)  **Primary key** clientNo |
| 5 | **ClientTel** (tel, clientNo)  **Primary key** tel  **Foreign key** clientNo references Client (clientNo) ON UPDATE CASCADE ON DELETE CASCADE |
| 6 | **Rent (**rentNo, rentStart, rentEnd, rentOver, isLate, duration, staffNo, clientNo, itemNo)  **Primary key** rentNo  **Foreign key** staffNo references Staff (staffNo) ON UPDATE CASCADE ON DELETE SET NULL  **Foreign key** clientNo references Client (clientNo) ON UPDATE CASCADE ON DELETE NO ACTION  **Foreign key** itemNo references Item (itemNo) ON UPDATE CASCADE ON DELETE NO ACTION |

4 SQL SCRIPT FOR IMPLEMENTING THE DATABASE

4.1 Creating tables

create table Staff

( staffNo varchar(5) not null primary key,

fname varchar(15) not null,

lname varchar(15) not null,

s\_position varchar(30) not null default 'Assistant',

sex char(1),

dob date

);

create table Item

( ISBN varchar(13) not null primary key,

item\_name varchar(30) not null,

author varchar(20),

publishing\_date date,

publishing\_house varchar(15),

bookshelfNo integer,

rackNo integer,

item\_state varchar(10)

);

create table ItemAuthor

( author varchar(20) not null primary key,

ISBN varchar(13) references item(ISBN) on update cascade on delete cascade

);

create table Client

( clientNo varchar(5) not null primary key,

fName varchar(15) not null,

lName varchar(15) not null,

telNo varchar(13) not null,

blacklisted char(1) not null default 'N',

debtValue integer not null default 0,

client\_class varchar(10) not null

);

create table ClientTel

( telNo varchar(13) not null primary key,

clientNo varchar(5) references Client(clientNo) on update cascade on delete cascade

);

create table Rent

( rentNo varchar(5) not null,

rentStart date not null,

rentEnd date not null,

rentOver date,

isLate char(1) not null default 'N',

clientNo varchar(5) references Client(clientNo) on update cascade on delete no action,

itemNo varchar(13) references Item(ISBN) on update cascade on delete no action

);

4.2 Populating tables

insert into staff values ('AA111', 'Judy', 'Jones', 'Manager', 'F', to\_date('26-06-1966', 'dd-MM-yyyy'));

insert into staff values ('AA112', 'Ryan', 'Jackson', 'Shelver', 'M', to\_date('13-09-1970', 'dd-MM-yyyy'));

insert into staff values ('AA113', 'Jimmy', 'Baker', 'Archivist', 'M', to\_date('20-04-1975', 'dd-MM-yyyy'));

insert into staff values ('AA114', 'Beverly', 'Coleman', 'Director', 'F', to\_date('13-03-1978', 'dd-MM-yyyy'));

insert into staff values ('AA115', 'Carolyn', 'Edwards', 'Assistant Director', 'F', to\_date('03-11-1988', 'dd-MM-yyyy'));

insert into staff values ('AA116', 'Linda', 'Powell', 'Information Assistant', 'F', to\_date('17-11-1965', 'dd-MM-yyyy'));

insert into staff values ('AA117', 'Martin', 'Foster', 'Stock Manager', 'M', to\_date('05-02-1966', 'dd-MM-yyyy'));

insert into staff values ('AA118', 'Nicole', 'Simmons', 'Clerk', 'F', to\_date('10-10-1968', 'dd-MM-yyyy'));

insert into staff values ('AA119', 'William', 'White', 'Supervisor', 'M', to\_date('25-04-1975', 'dd-MM-yyyy'));

insert into staff values ('AA120', 'Timothy', 'Flores', 'Assistant', 'M', to\_date('09-04-1985', 'dd-MM-yyyy'));

insert into item values ('9781234567897', 'Winter and the Preacher', 'Lauralee Knutson', to\_date('08-07-1910', 'dd-MM-yyyy'), 'Hachette', 1, 1, 'Available');

insert into item values ('9784965959849', 'Challenge the Savage', 'Myrtis Wingate', to\_date('10-08-1915', 'dd-MM-yyyy'), 'Hachette', 1, 2, 'Available');

insert into item values ('9788597966077', 'Exquisite Conditions', 'Nell Winkler', to\_date('11-01-1916', 'dd-MM-yyyy'), 'HarperCollins', 1, 3, 'Available');

insert into item values ('9788421931691', 'Battle Passage', 'Cindie Salcido', to\_date('11-11-1916', 'dd-MM-yyyy'), 'PRH', 1, 4, 'Available');

insert into item values ('9780588432004', 'Vault of the Whale', 'Augustine Quigley', to\_date('03-11-1925', 'dd-MM-yyyy'), 'PRH', 1, 5, 'Available');

insert into item values ('9782811381660', 'The Wall of Beauty', 'Lizabeth Terrell', to\_date('06-05-1928', 'dd-MM-yyyy'), 'Hachette', 1, 6, 'Available');

insert into item values ('9784962537149', 'Washington Scarlet', 'Brittani Pfeifer', to\_date('20-01-1915', 'dd-MM-yyyy'), 'PRH', 1, 7, 'Available');

insert into item values ('9786200748461', 'An Unsuitable Bride', 'Albina Maupin', to\_date('10-12-1917', 'dd-MM-yyyy'), 'HarperCollins', 1, 8, 'Available');

insert into item values ('9781844690985', 'Comfort and Mischief', 'Dell Geiger', to\_date('25-09-1939', 'dd-MM-yyyy'), 'Hachette', 1, 9, 'Available');

insert into item values ('9781184651462', 'The Promise of Guilt', 'Lovella Mcallister', to\_date('14-05-1946', 'dd-MM-yyyy'), 'S&S', 1, 10, 'Available');

insert into item values ('9781533069191', 'The Mountain Festival', 'Quinton Easley', to\_date('08-12-1947', 'dd-MM-yyyy'), 'S&S', 2, 1, 'Available');

insert into item values ('9782593086302', 'World of Tomorrow', 'Lourie Thiel', to\_date('16-02-1948', 'dd-MM-yyyy'), 'Hachette', 2, 2, 'Available');

insert into item values ('9787426045358', 'Situs Inversus', 'Brock Greenwood', to\_date('17-04-1951', 'dd-MM-yyyy'), 'S&S', 2, 3, 'Available');

insert into item values ('9787356368893', 'Distress of Hope', 'Thad Walden', to\_date('21-09-1955', 'dd-MM-yyyy'), 'HarperCollins', 2, 4, 'Available');

insert into item values ('9784638436196', 'Castle Hunters', 'Shantell Schreiber', to\_date('11-05-1969', 'dd-MM-yyyy'), 'Hachette', 2, 5, 'Available');

insert into item values ('9784609896004', 'Deception in Terra', 'Darius Applegate', to\_date('06-08-1973', 'dd-MM-yyyy'), 'S&S', 2, 6, 'Available');

insert into item values ('9787919256506', 'Velvet Agendas', 'Lavette Fisk', to\_date('16-12-1980', 'dd-MM-yyyy'), 'Hachette', 2, 7, 'Available');

insert into item values ('9782949301639', 'A House Built on Sand', 'Melynda Willard', to\_date('23-06-1986', 'dd-MM-yyyy'), 'S&S', 2, 8, 'Available');

insert into item values ('9784954047083', 'The Tale of the Assassin', 'Hosea Sherrod', to\_date('09-01-1988', 'dd-MM-yyyy'), 'PRH', 2, 9, 'Available');

insert into item values ('9787684817124', 'Born in the Precinct', 'Regena Willson', to\_date('15-03-1998', 'dd-MM-yyyy'), 'Hachette', 2, 10, 'Available');

insert into client values ('CA111', 'Norman', 'Schneider', '(530)476-7460', 'N', 0, 'Student');

insert into client values ('CA112', 'Tamara', 'Benson', '(303)563-1094', 'N', 0, 'Student');

insert into client values ('CA113', 'Vanessa', 'Wallace', '(712)226-2033', 'N', 0, 'Student');

insert into client values ('CA114', 'Kerry', 'Moran', '(423)515-7491', 'N', 0, 'Student');

insert into client values ('CA115', 'Trevor', 'Hogan', '(509)724-7406', 'N', 0, 'Student');

insert into client values ('CA116', 'Sherri', 'Higgins', '(234)307-4426', 'N', 0, 'Student');

insert into client values ('CA117', 'Lauren', 'Powell', '(727)860-5551', 'N', 0, 'Student');

insert into client values ('CA118', 'Rolando', 'Cross', '(651)430-8658', 'N', 0, 'Student');

insert into client values ('CA119', 'Jacquelyn', 'Tyler', '(408)854-8834', 'N', 0, 'Teacher');

insert into client values ('CA120', 'Dolores', 'Ferguson', '(646)232-9451', 'N', 0, 'Teacher');

insert into client values ('CA121', 'Eugene', 'Harrington', '(424)255-2115', 'N', 0, 'Student');

insert into client values ('CA122', 'Candace', 'Powers', '(269)769-2753', 'N', 0, 'Student');

insert into client values ('CA123', 'Clifton', 'Griffin', '(801)641-6470', 'N', 0, 'Teacher');

insert into client values ('CA124', 'Kent', 'Moran', '(563)288-8964', 'N', 0, 'Student');

insert into client values ('CA125', 'Jenny', 'Lindsey', '(509)956-8176', 'N', 0, 'Teacher');

insert into client values ('CA126', 'Maryann', 'Vasquez', '(706)509-2927', 'N', 0, 'Teacher');

insert into client values ('CA127', 'Guadalupe', 'Parks', '(469)643-1921', 'N', 0, 'Student');

insert into client values ('CA128', 'Maria', 'Rose', '(306)967-4716', 'N', 0, 'Student');

insert into client values ('CA129', 'Alma', 'Jimenez', '(304)557-7175', 'N', 0, 'Student');

insert into client values ('CA130', 'Francisco', 'Hodges', '(251)284-8217', 'N', 0, 'Student');

insert into rent values ('RA000', to\_date('21-06-2018', 'dd-MM-yyyy'), to\_date('21-09-2018', 'dd-MM-yyyy'), to\_date('20-09-2018', 'dd-MM-yyyy'), 'N', 'CA111', '9781234567897');

insert into rent values ('RA001', to\_date('05-03-2017', 'dd-MM-yyyy'), to\_date('05-06-2017', 'dd-MM-yyyy'), to\_date('04-06-2017', 'dd-MM-yyyy'), 'N', 'CA112', '9784965959849');

insert into rent values ('RA002', to\_date('06-02-2018', 'dd-MM-yyyy'), to\_date('06-05-2018', 'dd-MM-yyyy'), to\_date('05-05-2018', 'dd-MM-yyyy'), 'N', 'CA113', '9788597966077');

insert into rent values ('RA003', to\_date('07-05-2018', 'dd-MM-yyyy'), to\_date('07-08-2018', 'dd-MM-yyyy'), to\_date('06-08-2018', 'dd-MM-yyyy'), 'N', 'CA114', '9788421931691');

insert into rent values ('RA004', to\_date('31-01-2017', 'dd-MM-yyyy'), to\_date('01-05-2017', 'dd-MM-yyyy'), to\_date('30-04-2017', 'dd-MM-yyyy'), 'N', 'CA115', '9780588432004');

insert into rent values ('RA005', to\_date('15-07-2017', 'dd-MM-yyyy'), to\_date('15-10-2017', 'dd-MM-yyyy'), to\_date('14-10-2017', 'dd-MM-yyyy'), 'N', 'CA116', '9782811381660');

insert into rent values ('RA006', to\_date('09-02-2016', 'dd-MM-yyyy'), to\_date('09-05-2016', 'dd-MM-yyyy'), to\_date('08-05-2016', 'dd-MM-yyyy'), 'N', 'CA117', '9784962537149');

insert into rent values ('RA007', to\_date('20-03-2016', 'dd-MM-yyyy'), to\_date('20-06-2016', 'dd-MM-yyyy'), to\_date('19-06-2016', 'dd-MM-yyyy'), 'N', 'CA118', '9786200748461');

insert into rent values ('RA008', to\_date('24-04-2016', 'dd-MM-yyyy'), to\_date('24-07-2016', 'dd-MM-yyyy'), to\_date('23-07-2016', 'dd-MM-yyyy'), 'N', 'CA119', '9781844690985');

insert into rent values ('RA009', to\_date('01-05-2016', 'dd-MM-yyyy'), to\_date('01-08-2016', 'dd-MM-yyyy'), to\_date('31-07-2016', 'dd-MM-yyyy'), 'N', 'CA120', '9781184651462');

insert into rent values ('RA010', to\_date('22-05-2018', 'dd-MM-yyyy'), to\_date('22-08-2018', 'dd-MM-yyyy'), to\_date('21-08-2018', 'dd-MM-yyyy'), 'N', 'CA121', '9781533069191');

insert into rent values ('RA011', to\_date('13-06-2017', 'dd-MM-yyyy'), to\_date('13-09-2017', 'dd-MM-yyyy'), to\_date('12-09-2017', 'dd-MM-yyyy'), 'N', 'CA122', '9782593086302');

insert into rent values ('RA012', to\_date('06-07-2016', 'dd-MM-yyyy'), to\_date('06-10-2016', 'dd-MM-yyyy'), to\_date('05-10-2016', 'dd-MM-yyyy'), 'N', 'CA123', '9787426045358');

insert into rent values ('RA013', to\_date('21-11-2017', 'dd-MM-yyyy'), to\_date('21-02-2018', 'dd-MM-yyyy'), to\_date('20-02-2018', 'dd-MM-yyyy'), 'N', 'CA124', '9787356368893');

insert into rent values ('RA014', to\_date('25-02-2017', 'dd-MM-yyyy'), to\_date('25-05-2017', 'dd-MM-yyyy'), to\_date('24-05-2017', 'dd-MM-yyyy'), 'N', 'CA125', '9784638436196');

insert into rent values ('RA015', to\_date('27-01-2017', 'dd-MM-yyyy'), to\_date('27-04-2017', 'dd-MM-yyyy'), to\_date('26-04-2017', 'dd-MM-yyyy'), 'N', 'CA126', '9784609896004');

insert into rent values ('RA016', to\_date('08-06-2017', 'dd-MM-yyyy'), to\_date('08-09-2017', 'dd-MM-yyyy'), to\_date('07-09-2017', 'dd-MM-yyyy'), 'N', 'CA127', '9787919256506');

insert into rent values ('RA017', to\_date('27-07-2017', 'dd-MM-yyyy'), to\_date('27-10-2017', 'dd-MM-yyyy'), to\_date('26-10-2017', 'dd-MM-yyyy'), 'N', 'CA128', '9782949301639');

insert into rent values ('RA018', to\_date('17-08-2017', 'dd-MM-yyyy'), to\_date('17-11-2017', 'dd-MM-yyyy'), to\_date('16-11-2017', 'dd-MM-yyyy'), 'N', 'CA129', '9784954047083');

insert into rent values ('RA019', to\_date('01-01-2018', 'dd-MM-yyyy'), to\_date('01-04-2018', 'dd-MM-yyyy'), to\_date('31-03-2018', 'dd-MM-yyyy'), 'N', 'CA130', '9787684817124');

4.3 Simple queries

1. First and last names of directors of the library, the IDs of whom start with "AA".

select staffno, fname, lname, s\_position from staff where s\_position like '%Director' and staffno like 'AA%';

1. Staff IDs, first names, last names, positions and DOBs of all female library staff that were born in the 80's, listed as "Staff#", "First Name", "Last Name", "Current Position" and "Date of Birth".

select staffno "Staff#", fname "First Name", lname "Last Name", s\_position "Current Position", dob "Date of Birth" from staff where sex = 'F' and dob between '01-01-1980' and '31-12-1989';

1. Staff IDs, first and last names of male library managers, the IDs of whom start with "AA11".

select staffno, fname, lname from staff where staffno like 'AA11\_' and sex = 'M' and s\_position like '%Manager%';

1. ISBN, name, author and the publishing date of items in the library that have been published since 1990.

select isbn, item\_name, author, publishing\_date from item where publishing\_date between '01-01-1990' and current\_date;

1. ISBN, name and the author of items, names of which contain "the", that have been published by Hachette and are located in the first bookshelf of the library.

select isbn, item\_name, author from item where item\_name like '%the%' and publishing\_house = 'Hachette' and bookshelfno = 1;

1. Name, author, and publishing month of items that are located in bookshelf #2, ordered by publishing date and listed as "Item Name", "Author" and "Month Published".

select item\_name "Item Name", author "Author", to\_char(publishing\_date, 'Month') "Month Published" from item where bookshelfno = 2 order by 3;

1. ISBN, name, author listed as "ISBN", "Book", "Author" and location of items that have been published by S&S from 1910 to 1980.

select isbn "ISBN", item\_name "Book", author "Author", bookshelfno, rackno from item where publishing\_date between '01-01-1910' and '01-01-1980' and publishing\_house = 'S&S';

1. Client ID, first and last names and position of the teacher clients, the first names of whom begin with 'J'.

select clientno, fname, lname, client\_class from client where client\_class = 'Teacher' and fname like 'J%';

1. Client ID, first and last names and phone numbers of student clients, the IDs of whom start with "CA11", listed as "Client#", "Name", "Last Name" and"Phone Number", ordered by their first names.

select clientno "Client#", fname "Name", lname "Last Name", telno "Phone Number" from client where clientno like 'CA11\_' and client\_class = 'Student' order by 2;

1. Client ID, first and last names of clients that are not blacklisted, have no debts and contain letters 'l' and 'a' in their first names.

select clientno "Client #", lname "Last Name", fname "First Name" from client where blacklisted = 'N' and debtvalue = 0 and fname like '%l%' and fname like '%a%';

4.4 Queries with single-row functions

1. Item name, publishing date, and days since publishing of items shown as "Item", "Date Published" and "Days since publishing"

select item\_name "Item", publishing\_date "Date Published", trunc(current\_date - publishing\_date) "Days since publishing" from item;

1. Item name, length of the item name with spaces and the day of the week the item's been published, ordered by item names' length.

select item\_name, length(item\_name), to\_char(publishing\_date, 'fmDay') from item order by 2;

1. First name, last name and age of staff working in the library, ordered by age.

select lname, fname, to\_char(age(current\_date, dob), 'yy') from staff order by 3;

1. First name, last name and month of birth of staff working in the library, ordered by month.

select fname, lname, to\_char(dob, 'fmMonth') from staff order by 3;

1. First name, last name and position of the clients shown as "T" for teachers and "S" for students, shown as "First Name", "Last Name" and "Position", listing teachers first, and ordering alphabetically.

select fname "First Name", lname "Last Name", case client\_class

when 'Teacher' then 'T'

when 'Student' then 'S' end "Position"

from client order by 3 desc;

4.5 Queries from multiple tables

1. Display rent numbers with the clients.

select fname || ' ' || lname, rentno from rent natural join client

1. Displays rents with information on renter names and book names.

select r.rentno, c.fname || ' ' || c.lname, i.item\_name

from rent r join client c

on(c.clientno = r.clientno)

join item i

on(i.isbn = r.itemno)

1. The list of books that are not on the same bookshelf as the book "A House Built on Sand"

select i.item\_name, i.author

from item i join item a

on (a.item\_name = 'A House Built on Sand')

where i.bookshelfno <> a.bookshelfno

1. Shows the list of staff members that are younger than Jackson.

select s.fname, s.lname, s.dob

from staff s join staff j

on (j.lname = 'Jackson')

where s. dob > j.dob

1. List of clients that are not in the same class as Higgins and that are not blacklisted.

select c.clientno, c.fname, c.lname

from client c join client h

on (h.lname = 'Higgins')

where c.client\_class <> h.client\_class

and c.blacklisted = 'N'

4.6 Queries with aggregating data

1. Displays Minimal, maximal and average age of the staff.

select min(age(dob)) "Minimum age", max(age(dob)) "Maximum age", avg(age(dob)) "Average age" from staff

1. Average rent duration of not belated rents (days).

select round(avg(rentover - rentstart)) from rent where islate = 'N'

1. Amount of female directors that work in the library.

select count(sex) from staff where sex = 'F' and s\_position like '%Director'

4.7 Queries with subqueries

1. Displays the oldest book that is available and is published by Hachette.

select item\_name, author, age(publishing\_date) from item

where publishing\_house = 'Hachette' and item\_state = 'Available' and publishing\_date =

(select min(publishing\_date) from item)

1. Display the rent(s) with minimal duration period.

select rentno from rent where (rentover - rentstart) = (select min(rentover - rentstart) from rent)

1. Show up the youngest staff member of the library.

select fname, lname, dob from staff where age(dob) = (select min(age(dob)) from staff)