Dezhi, Fawwaz, Matthew, Tien

University Library System

ECS740P – Database Systems

Coursework 2

Group 27

Introduction

Relational Schema

Creating Tables

Views

Triggers

Data

Queries

Security Considerations

Introduction

This second report concerns the implementation of the conceptual database system outlined in our previous report. We will be using Oracle’s Live SQL environment to build and test all data. We will cover the creation of tables, views (virtual tables) and triggers, fill the system with test data and then run several queries to demonstrate possible use cases. Appropriate constraints will be implemented in our system in various ways to ensure data integrity, validity, and security. Our original conceptual schema is described below, primary keys are shown by underlined attributes and italicised attributes represent foreign keys.

**Subject** (Subject Code, Subject Title)

**Resource Details** (Catalogue Number, *Subject Code*, Maker, Publisher, Title, Publication Year, Format)

**Resource Location** (Location ID, Floor Number, Shelf Mark)

**Resource** (Resource ID, *Catalogue Number*, *Location ID*, Loan Period)

**Membership Constraints** (Member Class, Daily Fine Rate, Loan Limit)

**Member** (Library Card ID, *Member Class*, Full Name, Email, Status, Registration Date)

**Loan** (Loan ID, *Library Card ID*, *Resource ID*, Loan Start Date, Return Date)

**Fines Payment** (Payment ID, *Library Card ID*, Payment Date, Amount Paid)

Relational Schema

Upon reflection, we decided that our original schema needed to be amended slightly before proceeding with the implementation stage. Entities containing the word “Resource” have been changed to include the term “Item”, as “Resource” is a reserved keyword in Oracle. The only significant changes are to the Fines Payment table. The primary key has been changed to a composite key composed of 3 foreign keys, and “Amount Paid” was changed to “Total Paid”. The updated schema is reflected in the description below. This is the exact format that will be used to actually construct the system. Attributes that are both underlined and in italics are foreign primary keys.

**Subject** (Subject Code, Subject Title)

**Item Details** (Catalogue Number, *Subject Code*, Maker, Publisher, Title, Publication Year, Format)

**Item Location** (Location ID, Floor Number, Shelf Mark)

**Item** (*Item ID*, *Catalogue Number*, *Location ID*, Loan Period)

**Membership Constraints** (Member Class, Daily Fine Rate, Loan Limit)

**Member** (Library Card ID, *Member Class*, Full Name, Email, Status, Registration Date)

**Loan** (Loan ID, *Library Card ID*, *Item ID*, Loan Start Date, Return Date)

**Fines Payment** (*Loan ID*, *Library Card ID*, *Item ID*, Payment Date, Total Paid)

Creating Tables

We have 8 tables to create and they have to be structured and created in a specific sequence and this means they have to run in a specific order. This is because particular relations possess foreign keys and coding for a foreign key column before its table exists crashes the code, the solution was to create these tables beforehand.

The reason we used the VARCHAR2 datatype instead of VARCHAR is because although VARCHAR is defined in the ANSI standard, Oracle’s implementation of VARCHAR is actually not ANSI-compliant because it considers an empty string to be equivalent to NULL. Oracle may change this in the future thereby breaking legacy compatibility. However VARCHAR2 is functionally identical and will never be changed thus it is preferable to use it to ensure the functionality of the system does not change over time.

Subject Table

Within the subject table, subject\_code is given the primary key constraint. Therefore, all values within this column are both unique and not null. The only other attribute that accompanies subject\_code is subject\_title. Subject\_title is also constrained by being not null. It is paramount the information of subject\_title corresponds to the correct subject code – this was achieved. The primary key subject\_code would later be used a foreign key for another relation: item\_details.

CREATE TABLE subject (

subject\_code NUMBER(5) NOT NULL,

subject\_title VARCHAR2(255) NOT NULL,

CONSTRAINT pk\_subcode PRIMARY KEY (subject\_code)

);

Item Location Table

Within the item\_location table, location\_id is given the primary key constraint. Therefore, all values within this column are both unique and not null. The other attributes that accompany location\_id are floor\_number and shelf\_mark. Floor\_number and shelf\_mark are also constrained by being not null. In addition to this, floor\_number is also constrained with a CHECK, this check is to ensure the number of integers are 1 digit long and only 1 digit long. This was fulfilled through the line of code: CONSTRAINT chk\_floornum CHECK (LENGTH(floor\_number) =1. The purpose of this table was to ultimately create a primary key, location\_id, to become a foreign key for the item relation table.

CREATE TABLE item\_location (

location\_id NUMBER(10) NOT NULL,

floor\_number NUMBER(1) NOT NULL,

shelf\_mark NUMBER(10) NOT NULL,

CONSTRAINT pk\_loc PRIMARY KEY (location\_id),

CONSTRAINT chk\_floornum CHECK (LENGTH(floor\_number) = 1)

);

Item Details Table

Within the item\_details table, the attribute catalogue\_number is given the primary key constraint. Therefore, all values within this column are unique, not null and constrained to a mandatory 10 digit combination. The other attributes that accompany catalogue\_number are: subject\_code, maker, publisher, title, publication\_year and format\_type. As mentioned before subject\_code is constrained as a foreign key in this table to reference the subject table and is constrained with not null. The practice of using subject\_code as a foreign key successfully ties together a catalogue\_number with the subject it regards. The other attributes are also constrained with not null and further constraints are given to format\_type - an item in this relation can exist only as: ‘BOOK’, ‘CD’, ‘DVD’ or ‘VHS’. The final attribute with a constraint is publication\_year with a mandatory length of 4 digits to detail the year of the publication for the item. Only one instance of a catalogue number is recorded within this table and it is not possible to track each copy of an item through this table on its own. The purpose is solely for information about the items held within the library.

CREATE TABLE item\_details (

catalogue\_number NUMBER(10) NOT NULL,

subject\_code NUMBER(5) NOT NULL,

maker VARCHAR2(255) NOT NULL,

publisher VARCHAR2(255) NOT NULL,

title VARCHAR2(255) NOT NULL,

publication\_year NUMBER(4) NOT NULL,

format\_type VARCHAR2(255) NOT NULL,

CONSTRAINT pk\_catnum PRIMARY KEY (catalogue\_number),

CONSTRAINT fk\_subcode FOREIGN KEY (subject\_code)

REFERENCES subject (subject\_code),

CONSTRAINT chk\_catnum CHECK (LENGTH(catalogue\_number) = 10),

CONSTRAINT chk\_format\_type CHECK (format\_type IN ('BOOK', 'CD', 'DVD', 'VHS')),

CONSTRAINT chk\_pubyear CHECK (LENGTH(publication\_year) = 4)

);

Item Table

Within the item table, the attribute item\_id is given the primary key constraint. Therefore, all values within this column are both unique and not null. Catalogue\_number is a foreign key and references the item\_details table. Location\_id is also a foreign key and references the item\_location table. Loan\_period is also constrained with a check where number values within this column loan\_period must be either 14, 2 or 0. All attributes in this table are constrained with not null.

CREATE TABLE item (

item\_id NUMBER(10) NOT NULL,

catalogue\_number NUMBER(10) NOT NULL,

location\_id NUMBER(10) NOT NULL,

loan\_period NUMBER(2) NOT NULL,

CONSTRAINT pk\_itemid PRIMARY KEY (item\_id),

CONSTRAINT fk\_catnum FOREIGN KEY (catalogue\_number)

REFERENCES item\_details (catalogue\_number),

CONSTRAINT fk\_locid FOREIGN KEY (location\_id)

REFERENCES item\_location (location\_id),

CONSTRAINT chk\_loanperiod CHECK (loan\_period IN (14, 2, 0))

);

Member Constraint Table

Within this table, member\_class is given the primary key constraint. Therefore, all values within this column are both unique, not null and are constrained to have string values ‘STUDENT’ or ‘STAFF’. As member of the college library can only be a student or staff member. All the attributes within this table are constrained with not null. Loan\_limit is further constrained with a CHECK, this check is to ensure that only digits 5 and 10 are within the loan\_limit column. The purpose of member\_class primary key is to become a foreign key for the members table. Daily\_fine\_rate details the fine rate owed to the library if a member has not returned an item by the due date.

CREATE TABLE member\_constraints (

member\_class VARCHAR2(255) NOT NULL,

daily\_fine\_rate NUMBER(2) NOT NULL,

loan\_limit NUMBER(2) NOT NULL,

CONSTRAINT pk\_memberclass PRIMARY KEY (member\_class),

CONSTRAINT chk\_memberclass CHECK (member\_class IN ('STUDENT', 'STAFF')),

CONSTRAINT chk\_loanlimit CHECK (loan\_limit IN (5, 10))

);

Members Table

Within this table, library\_card\_id is given the primary key constraint. Therefore, all values within this column are both unique and not null. Member\_class is a foreign key which references the table Member Constraints and is also not null. All other attributes within this table are constrained to being not null. Additional CHECK constraints are given to the attributes status and email. For status, the check ensures the string values are either ‘ACTIVE’ or ‘SUSPENDED’. For email the check ensures the email contains an @ and a full stop to ensure emails are entered correctly. Having said this, if a user presses a full stop before the @ there is no system to prevent this from being entered, perhaps more specificity is needed here.

CREATE TABLE members (

library\_card\_id NUMBER(10) NOT NULL,

member\_class VARCHAR2(255) NOT NULL,

full\_name VARCHAR2(255) NOT NULL,

email VARCHAR2(255) NOT NULL,

status VARCHAR2(255) NOT NULL,

registration\_date DATE NOT NULL,

CONSTRAINT pk\_libcardid PRIMARY KEY (library\_card\_id),

CONSTRAINT fk\_memberclass FOREIGN KEY (member\_class)

REFERENCES member\_constraints (member\_class),

CONSTRAINT chk\_status CHECK (status IN ('ACTIVE', 'SUSPENDED')),

CONSTRAINT chk\_email CHECK (email LIKE '%@%'),

CONSTRAINT chk\_email2 CHECK (email LIKE '%.%')

);

Loan Table

Within this table, loan\_id is given the primary key constraint. Therefore, all values within this column are both unique and not null. Library\_card\_id is a foreign key and references the members table. Item\_id is also a foreign key and references the item table. All attributes within this table are constrained to being not null. An additional CHECK constraint is added between loan\_start\_date and return\_date attributes to prevent an impossibility that a loan can have a later date than the return date. This was achieved through this line of code: CONSTRAINT chk\_loandate\_returndate CHECK (loan\_start\_date < return\_date).

CREATE TABLE loan (

loan\_id NUMBER(10) NOT NULL,

library\_card\_id NUMBER(10) NOT NULL,

item\_id NUMBER(10) NOT NULL,

loan\_start\_date DATE NOT NULL,

return\_date DATE,

CONSTRAINT pk\_loanid PRIMARY KEY (loan\_id),

CONSTRAINT fk\_libcardid FOREIGN KEY (library\_card\_id)

REFERENCES members (library\_card\_id),

CONSTRAINT fk\_itemid FOREIGN KEY (item\_id)

REFERENCES item (item\_id),

CONSTRAINT chk\_loandate\_returndate CHECK (loan\_start\_date < return\_date)

);

Fines Payment Table

Within this table, loan\_id, library\_card\_id and item\_id are given the primary key constraint. It was decided to use a composite primary key for this table to provide specific detail of members of the college library who have paid fines and to identify them. The other attributes payment\_date and total\_paid are constrained as not null.

CREATE TABLE fines\_payment (

loan\_id NUMBER(10) REFERENCES loan (loan\_id) NOT NULL,

library\_card\_id NUMBER(10) REFERENCES members (library\_card\_id) NOT NULL,

item\_id NUMBER(10) REFERENCES item (item\_id),

payment\_date DATE NOT NULL,

total\_paid NUMBER(10) NOT NULL,

PRIMARY KEY (loan\_id, library\_card\_id, item\_id)

);

Views

One way of improving the functionality of our system is through the implementation of views (virtual tables). Views allow us to represent complex queries to the user in a more coherent manner that abstracts away the complexity of the database design. This section outlines 4 views which we implemented.

**Popular Items**

Now to create this view to discover popular loans required two JOINS. The initial INNER JOIN was between tables loan and item on matching values in attributes for item\_id. The second INNER JOIN was between item\_details and item to match values in attributes for catalogue\_number. The WHERE clause is used to specify loans that have been returned and then columns are grouped by item\_details.catalogue\_number and item\_details.title, then ordered from highest to lowest frequency to clearly identify the most popular title that has been on loan and returned. This view would be useful for the library admins to identify popular titles, and through using this VIEW in SQL Queries they could identify the most popular subjects and format types too. However, in this case the title of the items also details the format type and subject and so it is possible to identify the most popular format types and subject from this VIEW alone as the screenshots indicate.

CREATE VIEW popular\_item AS

SELECT COUNT(item\_details.catalogue\_number) AS highest\_frequency, item\_details.catalogue\_number, item\_details.title

FROM loan

INNER JOIN item ON item.item\_id = loan.item\_id

INNER JOIN item\_details ON item\_details.catalogue\_number = item.catalogue\_number

WHERE return\_date IS NOT NULL

GROUP BY item\_details.catalogue\_number, item\_details.title

ORDER BY highest\_frequency DESC;

Table

Description automatically generatedSELECT \* FROM popular\_item;

**Current Loans**

This VIEW is similar to the one above. This VIEW was created through an INNER JOIN on matching attributes item\_id from tables loan and item. Choice of the set of attributes was pivotal to make this VIEW informative to admins of the library by showing what specific catalogue\_numbers are on loan and who has that item on loan. Through an INNER JOIN on item\_id from both loan and item tables it was possible to match which items are on loan. Additionally, a WHERE clause was added to exclude items that have been returned to the library and so only data of items still on-loan were extracted through the script line: WHERE loan.return\_date IS NULL; Users will be able to identify which catalogue numbers are still on loan and when to expect their return through adding loan\_period to the loan\_start\_date.

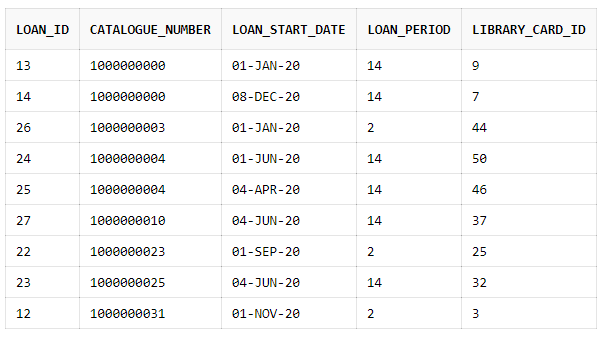
CREATE VIEW current\_loans AS

SELECT loan.loan\_id, item.catalogue\_number, loan.loan\_start\_date, item.loan\_period, loan.library\_card\_id

FROM loan

INNER JOIN item ON loan.item\_id = item.item\_id

WHERE loan.return\_date IS NULL;

**Current Suspensions**

This VIEW was created with the joining of three tables: loan, item and members. The purpose of this view is to automate data from three tables. Members of the library are suspended due to withholding a loan past its loan period and accruing overdue fines totalling a number greater than 10, then they are found in this table. This is achieved by initially matching item\_id values between the item and loan tables through an INNER JOIN. With the results, a LEFT JOIN is used with the members table on matching library\_card\_id attributes between members and loan table. A WHERE clause is then used on attributes return\_date (which is set equal to: IS NULL) and members.status (equated to: ‘SUSPENDED’) and (ROUND(SYSDATE - (item.loan\_period + loan.loan\_start\_date), 2)) > 10 detailing overdue loans and fines exceeding the £10. This VIEW is useful for admins of the library to identify members who are still due to pay their fines and return loans they have.

CREATE VIEW current\_suspensions AS

SELECT loan.loan\_id, loan.item\_id, loan.library\_card\_id, loan.loan\_start\_date, members.member\_class, members.full\_name, members.email,

members.registration\_date

FROM loan

INNER JOIN item

ON item.item\_id = loan.item\_id

LEFT JOIN members ON members.library\_card\_id = loan.library\_card\_id

WHERE return\_date IS NULL AND members.status = 'SUSPENDED' AND (ROUND(SYSDATE - (item.loan\_period + loan.loan\_start\_date), 2)) > 10

ORDER BY loan\_id ASC;

Table

Description automatically generatedSELECT \* FROM current\_suspensions;

**Fines Due**

For this VIEW, tables loan and item were joined with a select few attributes chosen from them. The aim of this view is to correctly calculate the fine amount due from members who have withheld items and incurred a fine. This includes members who are either suspended or just owe a fine on its own. This VIEW will be useful for library admins to see how much is owed to the library and members to see how much they owe to the library. Additionally, the code uses a ROUND function to 2 decimal places and SYSDATE which has a metric that goes right to the second, as a result, amount due increases as time progresses. Each day results in a £1 increase. This is seen as a huge success of the code and reaches the specification expectation. (Amount\_Due values will be different as time has progressed)

CREATE VIEW fines\_due AS

SELECT loan\_id, loan.library\_card\_id, loan.item\_id, (item.loan\_period + loan.loan\_start\_date) AS return\_deadline,

(ROUND(SYSDATE - (item.loan\_period + loan.loan\_start\_date), 2)) AS amount\_due

FROM loan

INNER JOIN item

ON item.item\_id = loan.loan\_id

Table

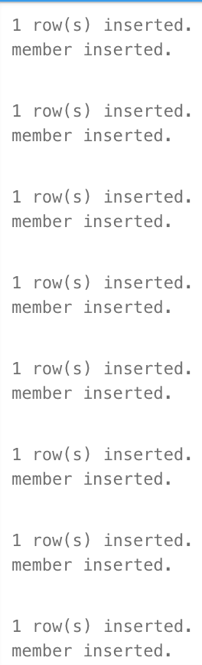
Description automatically generatedWHERE return\_date IS NULL;

SELECT \* FROM fines\_due;

(This line of code was included to show there’s no mistake in our understanding that ORACLE will not display the table without the script line: SELECT \* FROM fines\_due;)

Triggers

With regards to trigger implementation, BEFORE row triggers are more computationally efficient than AFTER row triggers because “with AFTER row triggers, affected data blocks must be read (logical read, not physical read) once for the trigger and then again for the triggering statement. Alternatively, with BEFORE row triggers, the data blocks must be read only once for both the triggering statement and the trigger.” (Oracle documentation).

**Member Entry**

CREATE OR REPLACE TRIGGER member\_entry

BEFORE INSERT ON members

BEGIN

DBMS\_OUTPUT.PUT\_LINE('member inserted.');

END;

/

This trigger is to indicate upon inserting data into the members table a system response of “member inserted” per insertion. A simple trigger that is to inform users that they have successfully inserted a new piece of data for the table member. To view the results as seen on the screenshot insert data into the member table – we have provided the data for this below, simply copy and paste the script and the results will be seen as the screenshot shows.

**A picture containing text

Description automatically generatedLoan Entry**

CREATE OR REPLACE TRIGGER loan\_insert

BEFORE INSERT ON loan

BEGIN

DBMS\_OUTPUT.PUT\_LINE('loan inserted.');

END;

/

This trigger is to indicate upon inserting data into the loan table a system response of “loan inserted” per insertion. A simple trigger that is to inform users that they have successfully inserted a new piece of data for the table loan. To view the results as seen on the screenshot insert data into the loan table – we have provided the data for this below, simply copy and paste the script and the results will be seen as the screenshot shows.

**Item Entry**

A picture containing text

Description automatically generatedCREATE OR REPLACE TRIGGER item\_insert

BEFORE INSERT ON item

BEGIN

DBMS\_OUTPUT.PUT\_LINE('item inserted.');

END;

/

This trigger is to indicate upon inserting data into the item table a System response of “item inserted” per insertion. A simple trigger that is to inform users that they have successfully inserted a new piece of data for the table item. To view the results as seen on the screenshot insert data into the item table – we have provided the data for this below, simply copy and paste the script and the results will be seen as the screenshot shows.

**Item Details Entry**

Graphical user interface, text

Description automatically generated

CREATE OR REPLACE TRIGGER item\_details\_insert

BEFORE INSERT ON item\_details

BEGIN

DBMS\_OUTPUT.PUT\_LINE('item details inserted.');

END;

/

This trigger is to indicate upon inserting data into item\_details table a System response of “item details inserted” per insertion. A simple trigger that is to inform users that they have successfully inserted a new piece of data for the table item\_details. To view the results as seen on the screenshot insert data into the item\_details table – we have provided the data for this below, simply copy and paste the script and the results will be seen as the screenshot shows.

Data

This section serves as a reference for all of the test data we entered into the system.

**Subject**

INSERT INTO subject (subject\_code, subject\_title) VALUES ('000', 'Computer Science, information and general works');

INSERT INTO subject (subject\_code, subject\_title) VALUES ('100', 'Philosophy and Psychology');

INSERT INTO subject (subject\_code, subject\_title) VALUES ('200', 'Religion');

INSERT INTO subject (subject\_code, subject\_title) VALUES ('300', 'Social Sciences');

INSERT INTO subject (subject\_code, subject\_title) VALUES ('400', 'Language');

INSERT INTO subject (subject\_code, subject\_title) VALUES ('500', 'Pure Science');

INSERT INTO subject (subject\_code, subject\_title) VALUES ('600', 'Technology');

INSERT INTO subject (subject\_code, subject\_title) VALUES ('700', 'Arts & Recreation');

INSERT INTO subject (subject\_code, subject\_title) VALUES ('800', 'Literature');

INSERT INTO subject (subject\_code, subject\_title) VALUES ('900', 'History & Geography');

**Item Location**

INSERT INTO item\_location (location\_id, floor\_number, shelf\_mark) VALUES (001, 1, 0001);

INSERT INTO item\_location (location\_id, floor\_number, shelf\_mark) VALUES (002, 1, 0002);

INSERT INTO item\_location (location\_id, floor\_number, shelf\_mark) VALUES (003, 1, 0003);

INSERT INTO item\_location (location\_id, floor\_number, shelf\_mark) VALUES (004, 1, 0004);

INSERT INTO item\_location (location\_id, floor\_number, shelf\_mark) VALUES (005, 2, 0005);

INSERT INTO item\_location (location\_id, floor\_number, shelf\_mark) VALUES (006, 2, 0006);

INSERT INTO item\_location (location\_id, floor\_number, shelf\_mark) VALUES (007, 2, 0007);

INSERT INTO item\_location (location\_id, floor\_number, shelf\_mark) VALUES (008, 3, 0008);

INSERT INTO item\_location (location\_id, floor\_number, shelf\_mark) VALUES (009, 3, 0009);

INSERT INTO item\_location (location\_id, floor\_number, shelf\_mark) VALUES (010, 3, 0010);

**Item Details**

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000000, 000, 'ClARK KENT', 'J.WILEY & SONS', 'THE BOOK OF COMPUTER SCIENCE, INFORMATION AND GENERAL WORKS', 1923, 'BOOK');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000001, 100, 'BRUCE WAYNE', 'PEARSON', 'THE BOOK OF PHILOSOPHY AND PSYCHOLOGY', 1919, 'BOOK');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000002, 200, 'HAL JORDAN', 'CAMBRIDGE', 'THE BOOK OF RELIGION', 2020, 'BOOK');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000003, 300, 'FRANKLIN RICHARDS', 'OXFORD', 'THE BOOK OF SOCIAL SCIENCES', 2018, 'BOOK');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000004, 400, 'ROBERTO MANCINI', 'MACMILLAN', 'THE BOOK OF LANGUAGE', 2016, 'BOOK');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000005, 500, 'ALBERT EINSTEIN', 'CENGAGE', 'THE BOOK OF PURE SCIENCE', 1924, 'BOOK');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000006, 600, 'ELON MUSK', 'CAMBRIDGE', 'THE BOOK OF TECHNOLOGY', 2003, 'BOOK');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000007, 700, 'JRR TOLKIEN', 'ALLEN & UNWIN', 'THE BOOK OF ARTS & RECREATION', 1955, 'BOOK');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000008, 800, 'WILLIAM SHAKESPEARE', 'J.WILEY & SONS', 'THE BOOK OF LITERATURE', 1576, 'BOOK');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000009, 900, 'MATTHEW MALLOY', 'MARVEL', 'THE BOOK OF HISTORY AND GEOGRAPHY', 2020, 'BOOK');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000010, 000, 'ClARK KENT', 'J.WILEY & SONS', 'THE CD OF COMPUTER SCIENCE, INFORMATION AND GENERAL WORKS', 2000, 'CD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000011, 100, 'BRUCE WAYNE', 'PEARSON', 'THE CD OF PHILOSOPHY AND PSYCHOLOGY', 1999, 'CD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000012, 200, 'HAL JORDAN', 'CAMBRIDGE', 'THE CD OF RELIGION', 2020, 'CD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000013, 300, 'FRANKLIN RICHARDS', 'OXFORD', 'THE CD OF SOCIAL SCIENCES', 2020, 'CD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000014, 400, 'ROBERTO MANCINI', 'MACMILLAN', 'THE CD OF LANGUAGE', 2018, 'CD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000015, 500, 'BILL NYE', 'CENGAGE', 'THE CD OF PURE SCIENCE', 1999, 'CD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000016, 600, 'ELON MUSK', 'CAMBRIDGE', 'THE CD OF TECHNOLOGY', 2010, 'CD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000017, 700, 'VIGGO MORTENSEN', 'ALLEN & UNWIN', 'THE CD OF ARTS & RECREATION', 2001, 'CD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000018, 800, 'BENJAMIN ZEPHANIAH', 'J.WILEY & SONS', 'THE CD OF LITERATURE', 2002, 'CD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000019, 900, 'MATTHEW MALLOY', 'MARVEL STUDIOS', 'THE CD OF HISTORY AND GEOGRAPHY', 2020, 'CD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000020, 000, 'MARK ZUCKERBERG', 'FACEBOOK', 'THE DVD OF COMPUTER SCIENCE, INFORMATION AND GENERAL WORKS', 2005, 'DVD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000021, 100, 'DESCARTES', 'PEARSON', 'THE DVD OF PHILOSOPHY AND PSYCHOLOGY', 2005, 'DVD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000022, 200, 'TOMMY THOMSON', 'CAMBRIDGE', 'THE DVD OF RELIGION', 2020, 'DVD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000023, 300, 'JIM JASPERS', 'OXFORD', 'THE DVD OF SOCIAL SCIENCES', 2020, 'DVD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000024, 400, 'CHRISTOPH WALTZ', 'MACMILLAN', 'THE DVD OF LANGUAGE', 2018, 'DVD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000025, 500, 'BILL NYE', 'CENGAGE', 'THE DVD OF PURE SCIENCE', 2001, 'DVD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000026, 600, 'KEANU REEVES', 'CAMBRIDGE', 'THE DVD OF TECHNOLOGY', 2010, 'DVD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000027, 700, 'VIGGO MORTENSEN', 'ALLEN & UNWIN', 'THE DVD OF ARTS & RECREATION', 2001, 'DVD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000028, 800, 'BENJAMIN ZEPHANIAH', 'J.WILEY & SONS', 'THE DVD OF LITERATURE', 2002, 'DVD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000029, 900, 'MATTHEW MALLOY', 'MARVEL STUDIOS', 'THE DVD OF HISTORY AND GEOGRAPHY', 2020, 'DVD');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000030, 000, 'JIM CARREY', 'PEARSON', 'THE VHS OF COMPUTER SCIENCE, INFORMATION AND GENERAL WORKS', 1995, 'VHS');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000031, 100, 'DESCARTES', 'PEARSON', 'THE VHS OF PHILOSOPHY AND PSYCHOLOGY', 1998, 'VHS');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000032, 200, 'TOMMY THOMSON', 'CAMBRIDGE', 'THE VHS OF RELIGION', 1980, 'VHS');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000033, 300, 'JIM JASPERS', 'OXFORD', 'THE VHS OF SOCIAL SCIENCES', 1990, 'VHS');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000034, 400, 'CHRISTOPH WALTZ', 'MACMILLAN', 'THE VHS OF LANGUAGE', 1999, 'VHS');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000035, 500, 'BILL NYE', 'CENGAGE', 'THE VHS OF PURE SCIENCE', 2001, 'VHS');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000036, 600, 'KEANU REEVES', 'CAMBRIDGE', 'THE VHS OF TECHNOLOGY', 1999, 'VHS');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000037, 700, 'VIGGO MORTENSEN', 'ALLEN & UNWIN', 'THE VHS OF ARTS & RECREATION', 2001, 'VHS');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000038, 800, 'BENJAMIN ZEPHANIAH', 'J.WILEY & SONS', 'THE VHS OF LITERATURE', 2002, 'VHS');

INSERT INTO item\_details (catalogue\_number, subject\_code, maker, publisher, title, publication\_year, format\_type) VALUES (1000000039, 900, 'MATTHEW MALLOY', 'MARVEL STUDIOS', 'THE VHS OF HISTORY AND GEOGRAPHY', 2002, 'VHS');

**Item**

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(1, 1000000000, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(2, 1000000000, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(3, 1000000000, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(4, 1000000000, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(5, 1000000000, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(6, 1000000001, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(7, 1000000001, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(8, 1000000001, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(9, 1000000001, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(10, 1000000001, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(11, 1000000002, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(12, 1000000002, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(13, 1000000002, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(14, 1000000002, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(15, 1000000002, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(16, 1000000003, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(17, 1000000003, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(18, 1000000003, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(19, 1000000003, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(20, 1000000003, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(21, 1000000004, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(22, 1000000004, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(23, 1000000004, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(24, 1000000004, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(25, 1000000004, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(26, 1000000005, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(27, 1000000005, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(28, 1000000005, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(29, 1000000005, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(30, 1000000005, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(31, 1000000006, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(32, 1000000006, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(33, 1000000006, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(34, 1000000006, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(35, 1000000006, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(36, 1000000007, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(37, 1000000007, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(38, 1000000007, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(39, 1000000007, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(40, 1000000007, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(41, 1000000008, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(42, 1000000008, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(43, 1000000008, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(44, 1000000008, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(45, 1000000008, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(46, 1000000009, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(47, 1000000009, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(48, 1000000009, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(49, 1000000009, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(50, 1000000009, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(51, 1000000010, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(52, 1000000010, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(53, 1000000010, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(54, 1000000010, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(55, 1000000010, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(56, 1000000011, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(57, 1000000011, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(58, 1000000011, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(59, 1000000011, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(60, 1000000011, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(61, 1000000012, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(62, 1000000012, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(63, 1000000012, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(64, 1000000012, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(65, 1000000012, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(66, 1000000013, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(67, 1000000013, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(68, 1000000013, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(69, 1000000013, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(70, 1000000013, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(71, 1000000014, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(72, 1000000014, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(73, 1000000014, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(74, 1000000014, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(75, 1000000014, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(76, 1000000015, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(77, 1000000015, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(78, 1000000015, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(79, 1000000015, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(80, 1000000015, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(81, 1000000016, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(82, 1000000016, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(83, 1000000016, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(84, 1000000016, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(85, 1000000016, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(86, 1000000017, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(87, 1000000017, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(88, 1000000017, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(89, 1000000017, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(90, 1000000017, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(91, 1000000018, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(92, 1000000018, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(93, 1000000018, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(94, 1000000018, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(95, 1000000018, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(96, 1000000019, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(97, 1000000019, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(98, 1000000019, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(99, 1000000019, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(100, 1000000019, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(101, 1000000020, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(102, 1000000020, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(103, 1000000020, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(104, 1000000020, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(105, 1000000020, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(106, 1000000021, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(107, 1000000021, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(108, 1000000021, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(109, 1000000021, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(110, 1000000021, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(111, 1000000022, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(112, 1000000022, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(113, 1000000022, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(114, 1000000022, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(115, 1000000022, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(116, 1000000023, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(117, 1000000023, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(118, 1000000023, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(119, 1000000023, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(120, 1000000023, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(121, 1000000024, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(122, 1000000024, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(123, 1000000024, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(124, 1000000024, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(125, 1000000024, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(126, 1000000025, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(127, 1000000025, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(128, 1000000025, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(129, 1000000025, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(130, 1000000025, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(131, 1000000026, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(132, 1000000026, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(133, 1000000026, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(134, 1000000026, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(135, 1000000026, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(136, 1000000027, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(137, 1000000027, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(138, 1000000027, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(139, 1000000027, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(140, 1000000027, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(141, 1000000028, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(142, 1000000028, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(143, 1000000028, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(144, 1000000028, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(145, 1000000028, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(146, 1000000029, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(147, 1000000029, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(148, 1000000029, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(149, 1000000029, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(150, 1000000029, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(151, 1000000030, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(152, 1000000030, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(153, 1000000030, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(154, 1000000030, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(155, 1000000030, 1, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(156, 1000000031, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(157, 1000000031, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(158, 1000000031, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(159, 1000000031, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(160, 1000000031, 2, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(161, 1000000032, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(162, 1000000032, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(163, 1000000032, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(164, 1000000032, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(165, 1000000032, 3, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(166, 1000000033, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(167, 1000000033, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(168, 1000000033, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(169, 1000000033, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(170, 1000000033, 4, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(171, 1000000034, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(172, 1000000034, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(173, 1000000034, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(174, 1000000034, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(175, 1000000034, 5, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(176, 1000000035, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(177, 1000000035, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(178, 1000000035, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(179, 1000000035, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(180, 1000000035, 6, 14);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(181, 1000000036, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(182, 1000000036, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(183, 1000000036, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(184, 1000000036, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(185, 1000000036, 7, 2);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(186, 1000000037, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(187, 1000000037, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(188, 1000000037, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(189, 1000000037, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(190, 1000000037, 8, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(191, 1000000038, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(192, 1000000038, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(193, 1000000038, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(194, 1000000038, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(195, 1000000038, 9, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(196, 1000000039, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(197, 1000000039, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(198, 1000000039, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(199, 1000000039, 10, 0);

INSERT INTO item (item\_id, catalogue\_number, location\_id, loan\_period) VALUES

(200, 1000000039, 10, 0);

**Member Constraints**

INSERT INTO member\_constraints (member\_class, daily\_fine\_rate, loan\_limit) VALUES

('STUDENT', 1, 5);

INSERT INTO member\_constraints (member\_class, daily\_fine\_rate, loan\_limit) VALUES ('STAFF', 1, 10);

**Members**

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000001, 'STAFF', 'Paul Turk', 'turkio@gmail.com', 'ACTIVE', TO\_DATE('12-01-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000002, 'STAFF', 'John Smith', 'hellojohn@yahoo.com', 'ACTIVE', TO\_DATE('15-01-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000003, 'STUDENT', 'Richard Howe', 'yolomami@gmail.com', 'SUSPENDED', TO\_DATE('10-05-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000004, 'STUDENT', 'Elliot Ellie', 'elliot@gmail.com', 'ACTIVE', TO\_DATE('04-08-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000005, 'STUDENT', 'Ellen Paige', 'epaige@yahoo.com', 'SUSPENDED', TO\_DATE('22-02-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000006, 'STAFF', 'Hellen Green', 'helllllen@gmail.com', 'ACTIVE', TO\_DATE('15-03-2019', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000007, 'STAFF', 'Jan Piskot', 'piskot@gmail.com', 'ACTIVE', TO\_DATE('04-09-2018', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000008, 'STAFF', 'Pascal Jouise', 'pascalito@yahoo.com', 'ACTIVE', TO\_DATE('11-11-2017', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000009, 'STUDENT', 'Anh Nguyen', 'anguyen@outlook.com', 'SUSPENDED', TO\_DATE('02-09-2019', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000010, 'STUDENT', 'Dominic Douche', 'ddouche@outlook.com', 'ACTIVE', TO\_DATE('14-01-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000011, 'STAFF', 'Valentina Napai', 'valentinan@gmail.com', 'ACTIVE', TO\_DATE('09-05-2019', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000012, 'STUDENT', 'Oscar Wilde', 'wildexo@yahoo.com', 'ACTIVE', TO\_DATE('16-09-2015', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000013, 'STUDENT', 'Isaac Newton', 'isaaaac@gmail.com', 'ACTIVE', TO\_DATE('04-04-2019', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000014, 'STAFF', 'Max Rex', 'maxrex@gmail.com', 'ACTIVE', TO\_DATE('15-02-2018', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000015, 'STUDENT', 'John Doe', 'johndoe@outlook.com', 'ACTIVE', TO\_DATE('11-04-2016', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000016, 'STUDENT', 'Sebastian Vettel', 'sebvettel@yahoo.com', 'ACTIVE', TO\_DATE('12-04-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000017, 'STUDENT', 'Adam Johnson', 'adamjohnson@gmail.com', 'ACTIVE', TO\_DATE('02-02-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000018, 'STUDENT', 'Luis Cavani', 'cavaniii@outlook.com', 'ACTIVE', TO\_DATE('05-09-2018', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000019, 'STUDENT', 'Vaclav Hasek', 'vachasek@outlook.com', 'ACTIVE', TO\_DATE('09-07-2016', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000020, 'STAFF', 'Paul Drake', 'pauldrake@gmail.com', 'ACTIVE', TO\_DATE('28-12-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000021, 'STUDENT', 'Aubrey Holzer', 'aubreyyy@gmail.com', 'ACTIVE', TO\_DATE('05-09-2015', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000022, 'STUDENT', 'Louise White', 'louiseee@outlook.com', 'ACTIVE', TO\_DATE('01-01-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000023, 'STUDENT', 'Whitney Heuston', 'wheus@yahoo.com', 'ACTIVE', TO\_DATE('15-11-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000024, 'STUDENT', 'Ollie Black', 'ollieee@gmail.com', 'ACTIVE', TO\_DATE('03-12-2019', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000025, 'STAFF', 'Anna Marie', 'annamarie@yahoo.com', 'SUSPENDED', TO\_DATE('11-10-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000026, 'STUDENT', 'Dua Kelzer', 'duakelzer@gmail.com', 'ACTIVE', TO\_DATE('02-02-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000027, 'STUDENT', 'Marie Chrome', 'chromemarie@gmail.com', 'ACTIVE', TO\_DATE('03-08-2018', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000028, 'STUDENT', 'Dominika Myslivcova', 'myslivcova@gmail.com', 'ACTIVE', TO\_DATE('05-08-2018', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000029, 'STUDENT', 'Irina Shaisal', 'irinakisskiss@outlook.com', 'ACTIVE', TO\_DATE('03-09-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000030, 'STUDENT', 'Russel Whitney', 'yoursrussel@gmail.com', 'ACTIVE', TO\_DATE('05-09-2018', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000031, 'STUDENT', 'Emil Yilka', 'emilpapi@gmail.com', 'ACTIVE', TO\_DATE('02-09-2018', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000032, 'STUDENT', 'Regina Dennis', 'reginaxoxxo@yahoo.com', 'SUSPENDED', TO\_DATE('04-09-2019', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000033, 'STAFF', 'Vincenzo Lombardi', 'lombardibombardi@gmail.com', 'SUSPENDED', TO\_DATE('05-07-2019', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000034, 'STUDENT', 'Bailey Suise', 'baileybaby@yahoo.com', 'ACTIVE', TO\_DATE('01-07-2015', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000035, 'STUDENT', 'Amelia Murat', 'ameliaaa@outlook.com', 'ACTIVE', TO\_DATE('14-01-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000036, 'STUDENT', 'Erica Mena', 'ericamena@gmail.com', 'ACTIVE', TO\_DATE('11-10-2019', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000037, 'STAFF', 'George Brocolli', 'brocolligeorge@yahoo.com', 'SUSPENDED', TO\_DATE('12-02-2018', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000038, 'STUDENT', 'Bruce Willis', 'bwill@gmail.com', 'ACTIVE', TO\_DATE('02-11-2018', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000039, 'STUDENT', 'Georgina Georg', 'ggbaby@gmail.com', 'ACTIVE', TO\_DATE('11-11-2016', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000040, 'STAFF', 'Margaret nat', 'margaretnat@gmail.com', 'ACTIVE', TO\_DATE('12-12-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000041, 'STUDENT', 'Maggie Lindsor', 'mslindsor@gmail.com', 'ACTIVE', TO\_DATE('12-04-2017', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000042, 'STUDENT', 'Charli Whittington', 'charliwhit@outlook.com', 'ACTIVE', TO\_DATE('11-05-2018', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000043, 'STUDENT', 'Margaret Thatchor', 'ironladie@yahoo.com', 'ACTIVE', TO\_DATE('04-12-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000044, 'STAFF', 'Freddie Singe', 'freddiesinge@yahoo.com', 'SUSPENDED', TO\_DATE('05-03-2019', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000045, 'STUDENT', 'Ozzy Parker', 'mrozzy@gmail.com', 'ACTIVE', TO\_DATE('09-11-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000046, 'STAFF', 'Anastasia Russia', 'letsgorussia@yahoo.com', 'SUSPENDED', TO\_DATE('07-05-2019', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000047, 'STUDENT', 'Dollorette Octavian', 'doloresbam@yahoo.com', 'ACTIVE', TO\_DATE('06-12-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000048, 'STAFF', 'Xerxes Austin', 'xxxxx@outlook.com', 'ACTIVE', TO\_DATE('05-11-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000049, 'STAFF', 'Harry Hughes', 'hhhhh@outlook.com', 'ACTIVE', TO\_DATE('11-03-2020', 'DD-MM-YYYY'));

INSERT INTO members (library\_card\_id, member\_class, full\_name, email, status, registration\_date) VALUES (0000000050, 'STUDENT', 'Leonardo Vinci', 'davinci@gmail.com', 'SUSPENDED', TO\_DATE('11-09-2019', 'DD-MM-YYYY'));

**Loans**

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(1, 1, 1, TO\_DATE('08-05-2020', 'DD-MM-YYYY'), TO\_DATE('10-05-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(2, 1, 35, TO\_DATE('08-05-2020', 'DD-MM-YYYY'), TO\_DATE('09-05-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(3, 1, 66, TO\_DATE('08-05-2020', 'DD-MM-YYYY'), TO\_DATE('10-05-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(4, 1, 75, TO\_DATE('08-05-2020', 'DD-MM-YYYY'), TO\_DATE('10-05-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(5, 1, 34, TO\_DATE('08-05-2020', 'DD-MM-YYYY'), TO\_DATE('10-05-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(6, 4, 2, TO\_DATE('10-12-2020', 'DD-MM-YYYY'), TO\_DATE('20-12-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(7, 4, 68, TO\_DATE('08-12-2020', 'DD-MM-YYYY'), TO\_DATE('10-12-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(8, 4, 73, TO\_DATE('08-12-2020', 'DD-MM-YYYY'), TO\_DATE('10-12-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(9, 10, 158, TO\_DATE('15-10-2020', 'DD-MM-YYYY'), TO\_DATE('16-10-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(10, 10, 181, TO\_DATE('15-10-2020', 'DD-MM-YYYY'), TO\_DATE('17-10-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(11, 19, 180, TO\_DATE('14-09-2018', 'DD-MM-YYYY'), TO\_DATE('20-09-2018', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(12, 3, 159, TO\_DATE('01-11-2020', 'DD-MM-YYYY'), NULL);

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(13, 9, 3, TO\_DATE('01-01-2020', 'DD-MM-YYYY'), NULL);

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(14, 7, 4, TO\_DATE('08-12-2020', 'DD-MM-YYYY'), NULL);

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(15, 16, 169, TO\_DATE('04-06-2020', 'DD-MM-YYYY'), TO\_DATE('08-06-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(16, 14, 56, TO\_DATE('01-06-2019', 'DD-MM-YYYY'), TO\_DATE('07-06-2019', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(17, 42, 60, TO\_DATE('01-01-2020', 'DD-MM-YYYY'), TO\_DATE('06-01-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(18, 28, 69, TO\_DATE('01-09-2020', 'DD-MM-YYYY'), TO\_DATE('08-09-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(19, 31, 85, TO\_DATE('04-06-2020', 'DD-MM-YYYY'), TO\_DATE('08-06-2020', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(20, 34, 132, TO\_DATE('01-06-2016', 'DD-MM-YYYY'), TO\_DATE('07-06-2016', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(21, 38, 157, TO\_DATE('04-04-2019', 'DD-MM-YYYY'), TO\_DATE('08-04-2019', 'DD-MM-YYYY'));

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(22, 25, 120, TO\_DATE('01-09-2020', 'DD-MM-YYYY'), NULL);

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(23, 32, 126, TO\_DATE('04-06-2020', 'DD-MM-YYYY'), NULL);

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(24, 50, 21, TO\_DATE('01-06-2020', 'DD-MM-YYYY'), NULL);

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(25, 46, 25, TO\_DATE('04-04-2020', 'DD-MM-YYYY'), NULL);

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(26, 44, 18, TO\_DATE('01-01-2020', 'DD-MM-YYYY'), NULL);

INSERT INTO loan (loan\_id, library\_card\_id, item\_id, loan\_start\_date, return\_date) VALUES

(27, 37, 53, TO\_DATE('04-06-2020', 'DD-MM-YYYY'), NULL);

**Fines Payment**

INSERT INTO fines\_payment (loan\_id, library\_card\_id, item\_id, payment\_date, total\_paid) VALUES (15, 16, 169, TO\_DATE('08-06-2020', 'DD-MM-YYYY'), 2);

INSERT INTO fines\_payment (loan\_id, library\_card\_id, item\_id, payment\_date, total\_paid) VALUES (16, 14, 56, TO\_DATE('07-06-2019', 'DD-MM-YYYY'), 4);

INSERT INTO fines\_payment (loan\_id, library\_card\_id, item\_id, payment\_date, total\_paid) VALUES (17, 42, 60, TO\_DATE('06-01-2020', 'DD-MM-YYYY'), 3);

INSERT INTO fines\_payment (loan\_id, library\_card\_id, item\_id, payment\_date, total\_paid) VALUES (18, 28, 69, TO\_DATE('08-09-2020', 'DD-MM-YYYY'), 5);

INSERT INTO fines\_payment (loan\_id, library\_card\_id, item\_id, payment\_date, total\_paid) VALUES (19, 31, 85, TO\_DATE('08-06-2020', 'DD-MM-YYYY'), 2);

INSERT INTO fines\_payment (loan\_id, library\_card\_id, item\_id, payment\_date, total\_paid) VALUES (20, 34, 132, TO\_DATE('07-06-2016', 'DD-MM-YYYY'), 4);

INSERT INTO fines\_payment (loan\_id, library\_card\_id, item\_id, payment\_date, total\_paid) VALUES (21, 38, 157, TO\_DATE('08-04-2019', 'DD-MM-YYYY'), 2);

Queries

The following scripts can be directly applied to this Library Database System on Oracle. Each query displayed in the screenshots will be explained in terms of its purpose and functionality for both members and library database managers. Annotated explanations may be used to highlight references to columns mentioned in the scripts.

Set of Queries to Identify Class Number (Subject Code) of Each item

This script joins two tables together to see which catalogue numbers are associated with what subject\_code - in addition, the attribute subject\_title is selected as well to clarify what the subject name is.

Query for Identifying Subject Code for Each Book Item

SELECT item\_id, item\_details.catalogue\_number, item\_details.title, item\_details.subject\_code, subject.subject\_title, publisher, publication\_year, format\_type

FROM item\_details

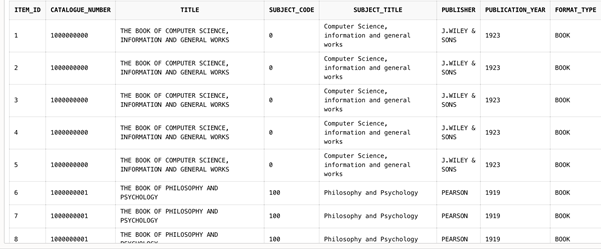
INNER JOIN subject

ON subject.subject\_code = item\_details.subject\_code

JOIN item

ON item\_details.catalogue\_number = item.catalogue\_number

WHERE format\_type = 'BOOK';



The script above returns each resource that is a ‘BOOK’ and returns a total of 50 items. The total of 50 items that are books is the entirety that the library contains of that type. There are 3 other format\_type available, each equalling the same total of 50 items per format\_type. The screenshots prove this, however, only the first 8 items which are books are shown here. The remaining 50 can be seen if you scroll down after inputting the Query script into Oracle SQL Live. This would be useful for library members attempting to find a book version of a particular title.

Query of Each Subject Code for Each CD Item

SELECT item\_id, item\_details.catalogue\_number, item\_details.title, item\_details.subject\_code, subject.subject\_title, publisher,

publication\_year, format\_type

FROM item\_details

INNER JOIN subject

ON subject.subject\_code = item\_details.subject\_code

JOIN item

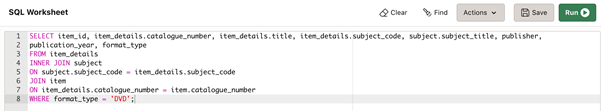
ON item\_details.catalogue\_number = item.catalogue\_number

Table

Description automatically generatedWHERE format\_type = 'CD';

The script above returns each resource that is a ‘CD’ and returns a total of 50 items. The total of 50 items that are CDs is the entirety that the library contains of that type. There are 2 other format\_type available, each equalling the same total of 50 items per format\_type. The screenshots prove this. Similarly to the previous script, members can easily find CD copies of an item through this query. The entire CD items can be viewed on Oracle – we had to save space as the screenshots took up a large bit of word document. This code works perfectly as displayed in the above screenshot.

Query for Each Subject Code of Each DVD Item



SELECT item\_id, item\_details.catalogue\_number, item\_details.title, item\_details.subject\_code, subject.subject\_title, publisher,

publication\_year, format\_type

FROM item\_details

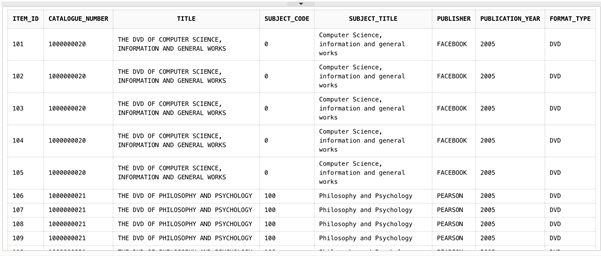
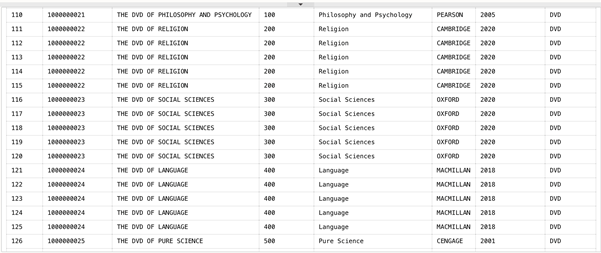
INNER JOIN subject

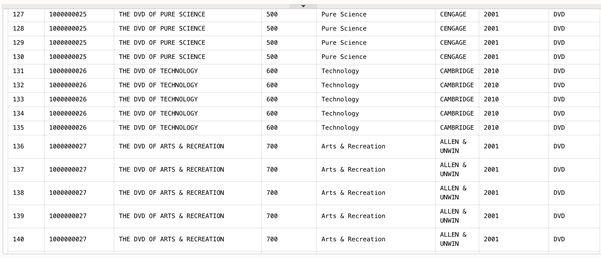
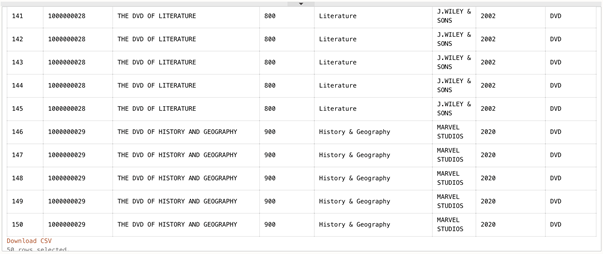
ON subject.subject\_code = item\_details.subject\_code

JOIN item

ON item\_details.catalogue\_number = item.catalogue\_number

WHERE format\_type = 'DVD';



The script above returns each resource that is a ‘DVD’ and returns a total of 50 items. The total of 50 items that are CDs is the entirety that the library contains of that type. There are 1 other format\_type available and equalling the same total of 50 items per format\_type. The screenshots prove this. Similarly to the previous script, members can easily find DVD copies of an item through this query.

Query for Each Subject Code for Each VHS Item



SELECT item\_id, item\_details.catalogue\_number, item\_details.title, item\_details.subject\_code, subject.subject\_title, publisher,

publication\_year, format\_type

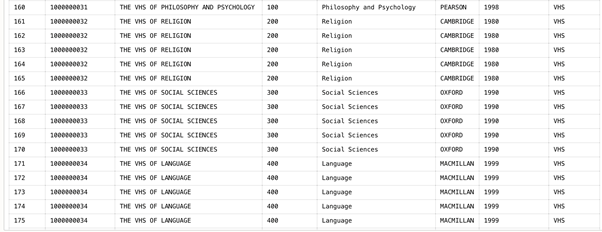
FROM item\_details

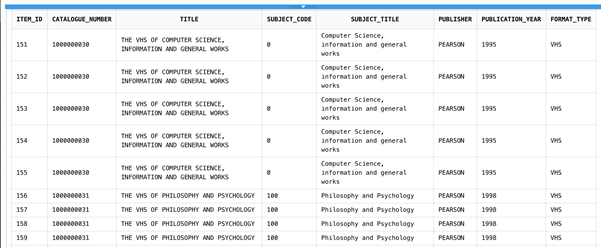
INNER JOIN subject

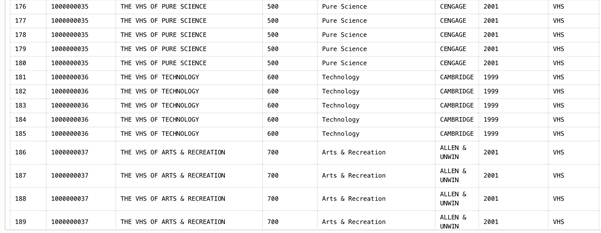
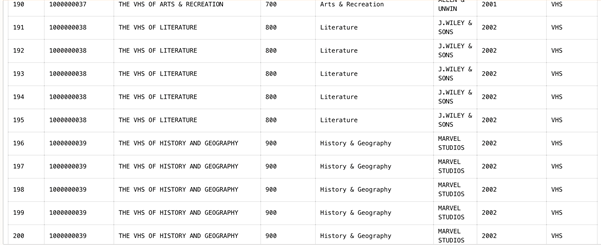
ON subject.subject\_code = item\_details.subject\_code

JOIN item

ON item\_details.catalogue\_number = item.catalogue\_number

WHERE format\_type = 'VHS';





Query Script to Count How Many Copies of Each Type of Item are Held by the Library

Query to Count Total Number of Books in Library

SELECT COUNT(\*) AS "Total Number of Books in Library" FROM (SELECT item\_id, item\_details.catalogue\_number, item\_details.title, item\_details.subject\_code, subject.subject\_title, publisher,

publication\_year, format\_type

FROM item\_details

INNER JOIN subject

ON subject.subject\_code = item\_details.subject\_code

JOIN item

ON item\_details.catalogue\_number = item.catalogue\_number

WHERE format\_type = 'BOOK');

The above script is specifically used to count the total number of Book items that exist in the Library when inventory is full and none of the items have been on loan. This query would be useful for admins to do a total inventory count for stock check for a particular type of item, in this case for books.

Graphical user interface, application

Description automatically generated

Query to Count Total Number of CDs in Library

SELECT COUNT(\*) AS "Total Number of CDs in Library" FROM (SELECT item\_id, item\_details.catalogue\_number, item\_details.title,

item\_details.subject\_code, subject.subject\_title, publisher,

publication\_year, format\_type

FROM item\_details

INNER JOIN subject

ON subject.subject\_code = item\_details.subject\_code

JOIN item

ON item\_details.catalogue\_number = item.catalogue\_number

Graphical user interface, application

Description automatically generatedWHERE format\_type = 'CD');



The above script is specifically used to count the total number of CD items that exist in the Library when inventory is full and none of the items have been on loan. This query would be useful for admins to do a total inventory count for stock check for a particular type of item, in this case for CDs.

Query to Count Total Number of DVDs in Library

SELECT COUNT(\*) AS "Total Number of DVDs in Library" FROM (SELECT item\_id, item\_details.catalogue\_number, item\_details.title,

item\_details.subject\_code, subject.subject\_title, publisher,

publication\_year, format\_type

FROM item\_details

INNER JOIN subject

ON subject.subject\_code = item\_details.subject\_code

JOIN item

ON item\_details.catalogue\_number = item.catalogue\_number

WHERE format\_type = 'DVD');

The above script is specifically used to count the total number of DVD items that exist in the Library when inventory is full and none of the items have been on loan. This query would be useful for admins and members to do a total inventory count for stock check for a particular type of item, in this case for DVDs.

Graphical user interface, text, application

Description automatically generated

Query to Count Total Number of VHSs in Library

SELECT COUNT(\*) AS "Total Number of VHS in Library" FROM (SELECT item\_id, item\_details.catalogue\_number, item\_details.title,

item\_details.subject\_code, subject.subject\_title, publisher,

publication\_year, format\_type

FROM item\_details

INNER JOIN subject

ON subject.subject\_code = item\_details.subject\_code

JOIN item

ON item\_details.catalogue\_number = item.catalogue\_number

WHERE format\_type = 'VHS');

Graphical user interface, text, application

Description automatically generatedThe above script is specifically used to count the total number of DVD items that exist in the Library when inventory is full and none of the items have been on loan. This query would be useful for admins and members to do a total inventory count for stock check for a particular type of item, in this case for VHSs.

Location of Each Resource

SELECT item\_id, item.catalogue\_number, item.location\_id, floor\_number, shelf\_mark,

subject\_code, title, format\_type

FROM item\_location

INNER JOIN item

ON item.location\_id = item\_location.location\_id

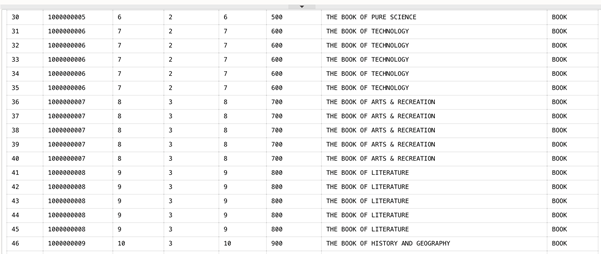
INNER JOIN item\_details

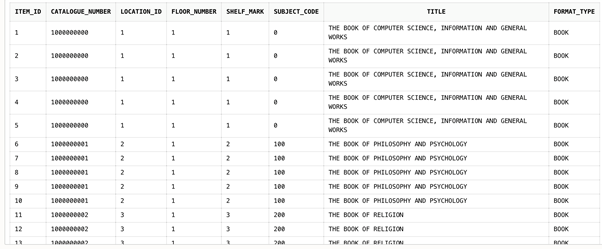
ON item.catalogue\_number = item\_details.catalogue\_number

ORDER BY item\_id ASC;

Table

Description automatically generatedThe code above provides the direct location of each resource item in the library. This query would be useful for library members seeking out particular books and they could use a simple WHERE clause to identify a specific title or a different format of the same title name. The information provided by the various attributes within this query also shows all format types that exist for a particular item along with its location. The screenshots are too long to fit in a word document, but the script is a guarantee that you will find all the items and their resource location.

Table

Description automatically generated

Query to Find Items with item\_id values above 50

SELECT item\_id, item.catalogue\_number, item.location\_id, floor\_number, shelf\_mark,subject\_code, title, format\_type

FROM item\_location

INNER JOIN item

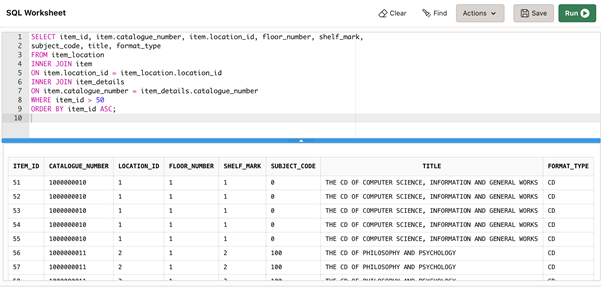
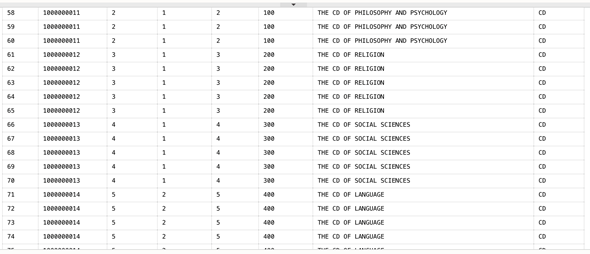
ON item.location\_id = item\_location.location\_id

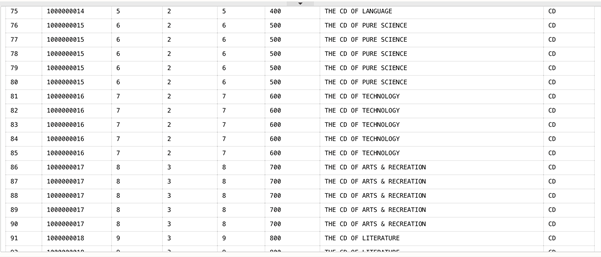
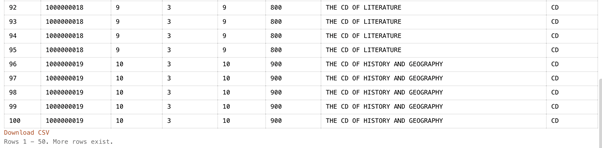
INNER JOIN item\_details

ON item.catalogue\_number = item\_details.catalogue\_number

WHERE item\_id > 50

ORDER BY item\_id ASC;

Same explanation as the previous script, difference here is merely the WHERE clause to find item\_id over 50 and ordered ascendingly.

****

**Query to find item\_id values above 100**

SELECT item\_id, item.catalogue\_number, item.location\_id, floor\_number, shelf\_mark,

subject\_code, title, format\_type

FROM item\_location

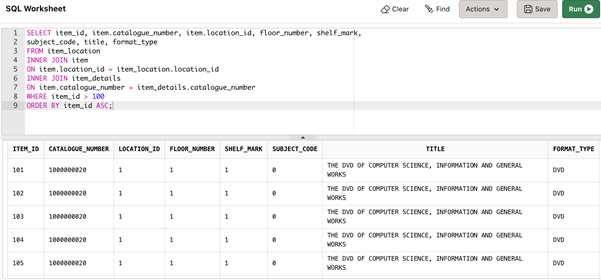
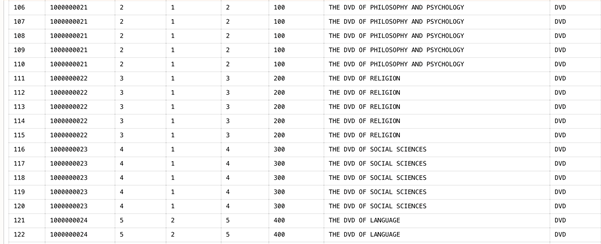
INNER JOIN item

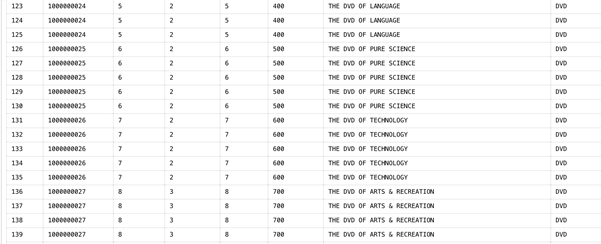
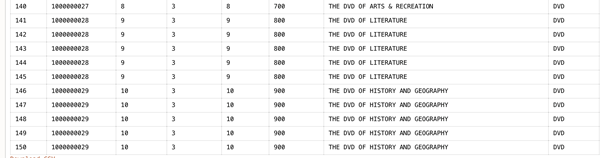
ON item.location\_id = item\_location.location\_id

INNER JOIN item\_details

ON item.catalogue\_number = item\_details.catalogue\_number

WHERE item\_id > 100

ORDER BY item\_id ASC;



**Query to Find item\_id values above 150**

SELECT item\_id, item.catalogue\_number, item.location\_id, floor\_number, shelf\_mark,

subject\_code, title, format\_type

FROM item\_location

INNER JOIN item

ON item.location\_id = item\_location.location\_id

INNER JOIN item\_details

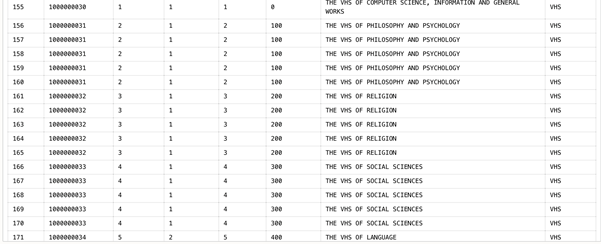
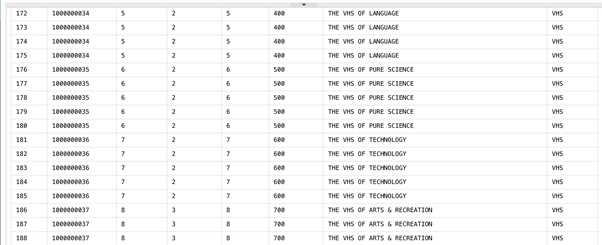
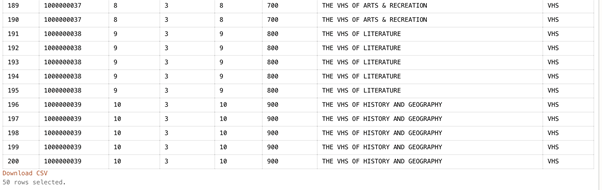
ON item.catalogue\_number = item\_details.catalogue\_number

WHERE item\_id > 150

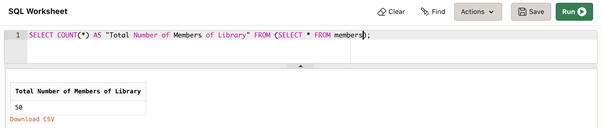
ORDER BY item\_id ASC;

Graphical user interface, application, table

Description automatically generated

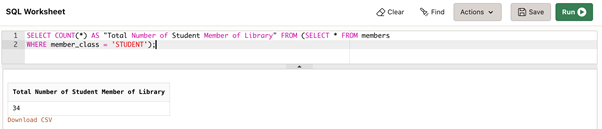


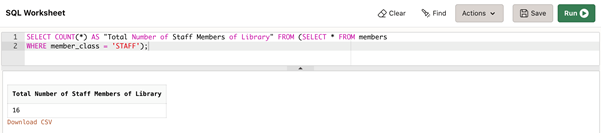
Query of Total Number of Members of the Library and Number of Students and Staff

SELECT COUNT(\*) AS "Total Number of Student Member of Library" FROM (SELECT \* FROM members);

SELECT COUNT(\*) AS "Total Number of Student Member of Library" FROM (SELECT \* FROM members

WHERE member\_class = 'STUDENT');

SELECT COUNT(\*) AS "Total Number of Student Member of Library" FROM (SELECT \* FROM members

WHERE member\_class = 'STAFF');

The query scripts above are useful for library admins to identify the total number of members in the library. This saves time from writing out complex queries.

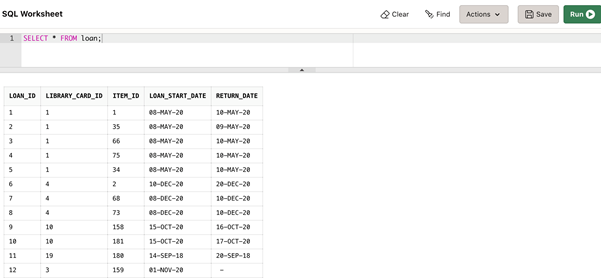
Query for Student and Staff member details

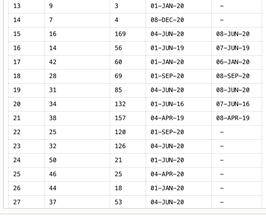
Graphical user interface, table

Description automatically generatedSELECT \* FROM members;

There are over 50 members in the library – to save space in the document only the first 15 have been screenshotted. However, this query yields all 50 members of the library. This query is useful as it displays all members and their contact details. This would be useful for a library admin if they needed to contact a member.

Query to View All Loans Held by Members Including If Loans have been Returned or Overdue

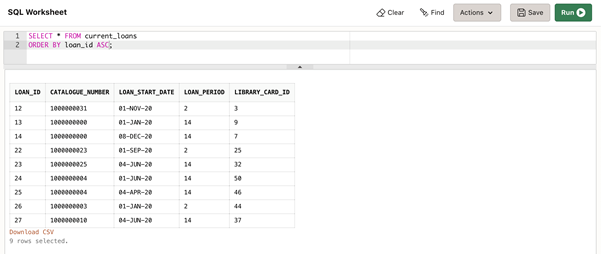
SELECT \* FROM loan;



The query script above show all loans both returned and currently on loan and is effective in detailing all the contents of the table loan. Library admins will find this useful.

Query to view Current Loans Exclusively

SELECT \* FROM current\_loans

ORDER BY loan\_id ASC;

The script above lists all the current loans available in the library. This query essentially shows the contents of the VIEW table current\_loans. Useful for admins to see what is on loan and useful for members to know who has which item and when they can expect it to be back in the library.

Record of Previous Loans

SELECT \* FROM popular\_item

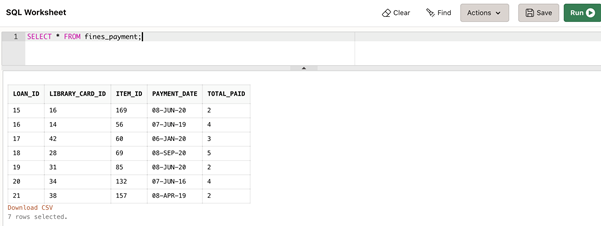
Table

Description automatically generated

This query is the contents of the VIEW popular\_item.

Query to Identify Fine Payment Details of Members

SELECT \* FROM fines\_payment;



This query highlights all the fines that have been paid. This is good for general accounting and tracking members that have a history of withholding loans longer than their loan period allows.

Query to Find Current Suspended Members of the Library

SELECT \* FROM current\_suspensions;

Table

Description automatically generatedThis query is essentially displaying the contents of the VIEW current\_suspensions and it is useful for identifying members of the library who have been suspended.

Security & Legal Considerations

The usage of parameterized queries would have allowed us to further constrain user input and lower the ability of hackers to utilize SQL injection attacks on our library system. Unfortunately, this feature does not work properly on Oracle SQL live so we were unable to implement any queries that used this technique.

As this is a UK based system, we must comply with the UK Data Protection Act 2018. One aspect is which is that personal data should be removed from the system once it is no longer necessary to keep it. In our system this would be when students graduate or when staff members leave the university.

Restricting data types where possible to only NUMBER or DATE ensures the appropriateness of data entered into the system, maintaining database integrity and utility. One of our views (current\_suspensions) exposes sensitive user data such as email addresses. However, in practice if our database application were to be deployed it would be password protected and any functionality including the querying of views would be limited to a protected administrator account. Furthermore, arbitrary queries are disallowed, with administrators being limited to the 12 pre-defined queries we have created. This removes any possibility of malicious actors compromising the integrity of our database system.