**Assignment Assessment Form (CLO2, CLO3)**

**Rating (Task 1, 2, 3, 4, 5 & 7) = 1: Very Poor, 2-3: Poor, 4-5: Average, 6-7: Good, 8-10: Excellent Rating (Task 6) = 1: Very Poor, 2: Poor, 3: Average, 4: Good, 5-6: Excellent**

**Task 6**

Form 3

Programme (Group): \_\_\_DCS1\_\_\_( 5 )

**Member Name**: 1. TAN KANG HONG , 2. LOW JUN YAN , 3. LAU JUN DIAN , 4. HAR CHUN WAI , 5.HO JING XIAN

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task**  **No.** | **Task Descriptions** | **Weightage** | **Criteria** | **1**  TAN KANG HONG | **2**  LOW JUN YAN | **3**  LAU JUN DIAN | **4**  HAR CHUN WAI | **5**  HO JING XIAN | **Comment** |
| 1 | Develop Business rules | 10% | 1. Include the required and relevant pairs of business rules. 2. All business rules must be clearly defined, precise, and reflect the policies and procedures of the organization’s operational environment. |  | | | | |  |
| 2 | Develop ERD | 10% | 1. Transform business rules to a relational database model correctly. 2. Correct use of Crow’s Foot notations. 3. Include all necessary entities, attribute & relationships. |  | | | | |  |
| 3 | Develop DBDL | 10% | 1. Correct use of DBDL format as required 2. All required entities, attributes and relationships correctly shown 3. Indicate Primary key and Foreign key clearly |  | | | | |  |
| 4 | Database Design  20% | 10% | 1. Correct tables, records and fields designed according to the ERD developed. |  | | | | |  |
| 10% | 1. Enforcement of entity integrity rule & referential integrity rule 2. Appropriate data types, default values and check constraints. |  | | | | |  |
| 5 | Records (Entries) | 10% | 1. Provide sufficient and quality data records 2. Well-designed records for adequate and logical choices of queries to be performed |  | | | | |  |
| 6 | Queries Design  30% | 10% | 1. Flexible query for variety of inputs. Clear & proper identification of information needs. 2. Apply Accept, Prompt and variable substitution in queries. Flexible query to cater for variety of inputs, use of multiple tables. 3. Apply Report Formatting features. Meaningful report handlings. Data values formatted accordingly. 4. **Only SELECT statements.** |  |  |  |  |  |  |
| 10% |  |  |  |  |  |  |
| 10% |  |  |  |  |  |  |
| 7 | Assignment Report | 10% | 1. Comprehensive, clarity and completeness coverage 2. Quality of report presented 3. Presentation and Q & A |  |  |  |  |  |  |
| **Assignment Marks / 100** | | |  |  |  |  |  |  |  |

**FORM 1**



**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY**

AACS3103 DATABASE DEVELOPMENT AND APPLICATIONS

Assignment

**Semester JAN 2021**

|  |  |  |
| --- | --- | --- |
| Programme | : | DCS |
| Tutorial Group | : | 5 |
| Date Submitted | : | 4-1-2021 |

Team members:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name (Block Letters)** | **Registration No.** | **Signature** | **Date** |
| 1 | TAN KANG HONG | 2002959 | TAN |  |
| 2 | LOW JUN YAN | 2003061 | YAN |  |
| 3 | LAU JUN DIAN | 2003133 | DIAN |  |
| 4 | HAR CHUN WAI | 2002982 | WAI |  |
| 5 | HO JING XIAN | 2002895 | XIAN |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Low Jun Yan | Tan Kang Hong | Lau Jun Dian | Har Chun Wai | Ho Jing Xian |

**FORM 2**



**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY**

**Plagiarism Statement and Guideline for Late Submission of Coursework**

Read, complete, and sign this statement to be submitted with the written report.

**We confirm that we have read and shall comply with all the terms and conditions of TAR University College’s plagiarism policy.**

**We declare that this assignment is free from all forms of plagiarism and for all intents and purposes is my own properly derived work.**

Declaration Statement Acknowledged by

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name (Block Letters)** | **Registration No.** | **Signature** | **Date** |
| 1 | TAN KANG HONG | 2002959 | TAN |  |
| 2 | LOW JUN YAN | 2003061 | YAN |  |
| 3 | LAU JUN DIAN | 2003133 | DIAN |  |
| 4 | HAR CHUN WAI | 2002982 | WAI |  |
| 5 | HO JING XIAN | 2002895 | XIAN |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Low Jun Yan | Tan Kang Hong | Lau Jun Dian | Har Chun Wai | Ho Jing Xian |

|  |  |
| --- | --- |
| **Table of Contents** | **PAGE** |
| Assignment Assessment Form | 1 |
| Declaration | 3 |
|  |  |
| **Task 1: Business Rules of the System** | 6 |
| 1.1 Business Rules and Assumptions |
|  |  |
| **Task 2: Entity-Relationship Diagram (ERD)** | 7 |
|  |  |
| **Task 3: Normalization** | 8 |
| 3.1 Attributes of entities with keys |
|  |  |
| **Task 4: Create Databases in Oracle** | 9 |
| 4.1 Branch Table |
| 4.2 Room Table |
| 4.3 Staff Table |
| 4.4 Book Table |
| 4.5 Student Table | 10 |
| 4.6 Book Loan Table |
| 4.7 Book Loan Details Table |
| 4.8 Room Booking Table | 11 |
| 4.9 Room Booking Details Table |
|  |  |
| **Task 5: Sample Data / Records** | 12 |
| 5.1 Student |
| 5.2 Branch |
| 5.3 Room | 13 |
| 5.4 Room Booking Details |
| 5.5 Book Loan |
| 5.6 Book Loan Details | 14 |
| 5.7 Book |
| 5.8 Staff | 15 |
| 5.9 Room booking |
|  |  |
| **Task 6: SQL Queries and Reports** | 16 |
| 6.1 Low Jun Yan |
| 6.1.1 Query 1: Detail report of student using library rooms in a range of date | 16-17 |
| 6.1.2 Query 2: Detail report of library rooms usage records | 18-19 |
| 6.1.3 Query 3: Detail report of staff with room use date | 20-21 |
| 6.2 Tan Kang Hong | 22 |
| 6.2.1 Query 1: Detail report of book with book loan ID | 22-23 |
| 6.2.2 Query 2: Detail report of book with start and end borrowed date | 23-24 |
| 6.2.3 Query 3: Detail report of book with search start and end due date | 25-27 |
| 6.3 Har Chun Wai | 28 |
| 6.3.1 Query 1: Detail report of book with borrow and return dates | 28-29 |
| 6.3.2 Query 2: Detail report of book loan details with grouped fine amount | 29-30 |
| 6.3.3 Query 3: Detail report of books borrowed by a student | 31-32 |
| 6.4 Ho Jing Xian | 33 |
| 6.4.1 Query 1: Detail report of books with book borrowed date | 33-34 |
| 6.4.2 Query 2: Detail report of student with book categories | 34-35 |
| 6.4.3 Query 3: Detail report of book borrowed within the range of student ID | 35-36 |
| 6.5 Lau Jun Dian | 37 |
| 6.5.1 Query 1: Detail report of the branch student booking which room | 37-38 |
| 6.5.2 Query 2: Detail report of staff handle room booking for branch students | 38-39 |
| 6.5.3 Query 3: Detail report of the branch student borrowed which book | 39-40 |

# Task 1: Business Rules of the System

## 1.1 Business Rules and Assumptions

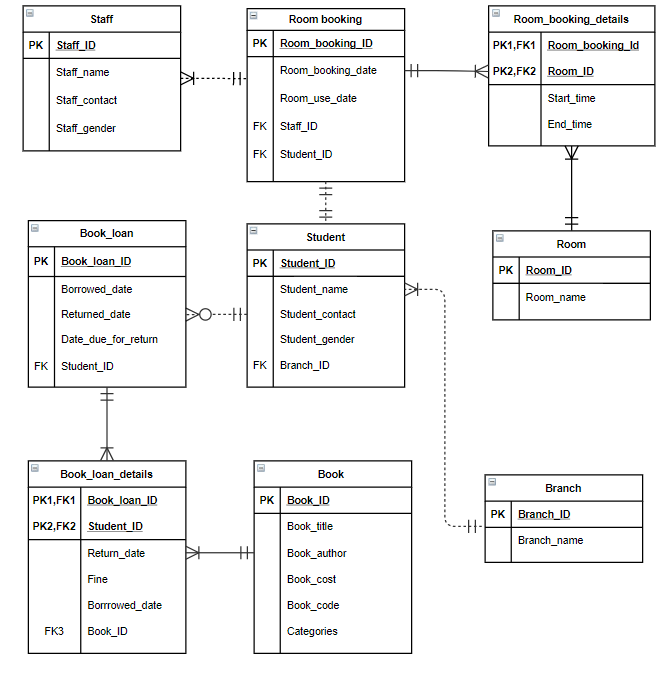
Business Rules:

1. Each student can have none or many book loans and each book loan can only be placed by one and only one student.
2. Each book loan can have one or many book loans details and each book loan details can be placed by one and only one book loan.
3. Each book loan details can have one and only one book and each book can be placed by one or many book loans details.
4. Each student can have one and only one room booking and each room booking can be placed by one and only one student.
5. Each room booking can have one or many room booking details and each room booking details can only be placed by one and only one room booking.
6. Each room booking details can have one and only one room and each room can be placed by one or many room booking details.
7. Each room booking can have one or many staff and each staff can only be placed by one and only one room booking.
8. Each student can have one and only one branch and each branch can be placed by one or many students.
9. Each student can borrow books, but must be returned within 4 weeks.
10. Do not use rooms in the library before 8am and after 8pm.

Assumptions:

1. Each student will at least borrow one book every month.
2. When an item is loaned the borrower’s details and book loan details are recorded.
3. Each student can only book one room at one time.

# Task 2: Entity-Relationship Diagram (ERD)



# Task 3: Normalization

## 3.1 Attributes of entities with keys

Student (Student\_ID, Student\_name, Student\_contact, Student\_gender, Branch\_ID\*)

Staff (Staff\_ID, Staff\_name, Staff\_contact, Staff\_gender,)

Branch (Branch\_ID, Branch\_name)

Room (Room\_ID, Room\_name)

Room\_booking (Room\_booking\_ID, Room\_booking\_date, Room\_use\_date, Staff\_ID\*, Student\_ID\*)

Room\_booking\_details (Room\_booking\_ID\*, Room\_ID\*, Start\_Time, End\_Time)

Book\_loan (Book\_loan\_ID, Borrowed\_date, Returned\_date, Date\_due\_for\_return, Student\_ID\*)

Book\_loan\_details (Book\_loan\_ID\*, Student\_ID, Borrowed\_date, Return\_date, Fine, Book\_ID\*)

Book (Book\_ID, Book\_title, Book\_author, Book\_cost, Book\_code, Categories)

# Task 4: Create Database in Oracle

## 4.1 Branch Table

CREATE TABLE Branch (

Branch\_ID VARCHAR(10) NOT NULL,

Branch\_name VARCHAR(50) NOT NULL,

PRIMARY KEY(Branch\_ID)

);

## 4.2 Room Table

CREATE TABLE Room (

Room\_ID varchar(8) NOT NULL,

Room\_name varchar(20) NOT NULL,

primary key(Room\_ID)

);

## 4.3 Staff Table

CREATE TABLE Staff (

Staff\_ID VARCHAR(4) NOT NULL

Staff\_name VARCHAR(50) NOT NULL,

Staff\_contact VARCHAR(14) NOT NULL,

Staff\_gender CHAR(1) DEFAULT 'M',

PRIMARY KEY(Staff\_ID),

CONSTRAINT chk\_staff\_gender check(UPPER(Staff\_gender) in ('M','F'))

);

## 4.4 Book Table

CREATE TABLE Book (

book\_ID varchar(7) NOT NULL,

book\_title varchar(30) NOT NULL,

book\_auther varchar(25) NOT NULL,

book\_cost number(7,2) NOT NULL,

book\_code varchar(20) NOT NULL,

categories varchar(15) DEFAULT 'All Type',

primary key(book\_ID)

);

## 4.5 Student Table

CREATE TABLE Student (

Student\_ID VARCHAR(10) NOT NULL,

Student\_name VARCHAR(50) NOT NULL,

Student\_contact VARCHAR(14) NOT NULL,

Student\_gender CHAR(1) DEFAULT 'M',

Branch\_ID VARCHAR(9) DEFAULT 'DCSY1S3',

PRIMARY KEY(Student\_ID),

FOREIGN KEY (Branch\_ID) references Branch (Branch\_ID),

CONSTRAINT chk\_student\_gender check(UPPER(Student\_gender) in ('M','F'))

);

## 4.6 Book Loan Table

CREATE TABLE Book\_Loan (

book\_loan\_ID varchar(9) NOT NULL,

borrowed\_date date NOT NULL,

returned\_date date NOT NULL,

date\_due\_for\_return date NOT NULL,

student\_ID varchar(10) NOT NULL,

PRIMARY KEY(book\_loan\_ID),

FOREIGN KEY(student\_ID) references student(student\_ID)

);

## 4.7 Book Loan Details Table

CREATE TABLE Book\_loan\_details (

book\_loan\_ID varchar(9) NOT NULL,

student\_ID varchar(10) NOT NULL,

borrowed\_date date NOT NULL,

return\_date date NOT NULL,

fine number(7,2) NOT NULL,

book\_ID varchar(7) NOT NULL,

primary key(book\_loan\_ID, student\_ID),

FOREIGN KEY (Student\_ID) references Student (Student\_ID),

foreign key(book\_loan\_ID) references Book\_loan(book\_loan\_ID),

foreign key(book\_ID) references Book(book\_ID)

);

## 4.8 Room Booking Table

CREATE TABLE Room\_booking (

Room\_booking\_ID VARCHAR(5) NOT NULL,

Room\_booking\_date date NOT NULL,

Room\_use\_date date NOT NULL,

Staff\_ID VARCHAR(4) NOT NULL,

Student\_ID VARCHAR(10) NOT NULL,

PRIMARY KEY(Room\_booking\_ID),

FOREIGN KEY(Staff\_ID) references Staff(Staff\_ID),

FOREIGN KEY(Student\_ID) references Student(Student\_ID)

);

## 4.9 Room Booking Details Table

CREATE TABLE Room\_booking\_details (

Room\_booking\_ID VARCHAR(5) NOT NULL,

Room\_ID VARCHAR(8) NOT NULL,

Start\_time CHAR(5),

End\_time CHAR(5),

PRIMARY KEY(Room\_booking\_ID, Room\_ID),

FOREIGN KEY (Room\_booking\_ID) references Room\_booking (Room\_booking\_ID),

FOREIGN KEY (Room\_ID) references Room (Room\_ID)

);

# Task 5: Create Records

## 5.1 Student

INSERT INTO Student VALUES ('20WMD02001','BAI JING TING','014-235-8478','M','DCSY1S3');

INSERT INTO Student VALUES ('20WMD02002','LEE RONG HAO','015-235-8478','M','DCSY1S3');

INSERT INTO Student VALUES ('20WMD02003','LAU JUN DIAN','016-235-8478','M','DCSY1S3');

INSERT INTO Student VALUES ('20WMD02004','LOW JUN YAN','017-235-8478','M','DCSY1S3');

INSERT INTO Student VALUES ('20WMD02005','HAR CHUN WAI','018-235-8478','M','DCSY1S3');

INSERT INTO Student VALUES ('20WMD02006','TENG LI QIAN','011-235-8478','F','DCSY1S3');

INSERT INTO Student VALUES ('20WMD02007','TONG HUN XIN','012-235-8478','F','DCSY1S3');

INSERT INTO Student VALUES ('20WMD02008','GOH YONG XUAN','013-235-8478','F','DCSY1S3');

INSERT INTO Student VALUES ('20WMD02009','TENG SI XIANG','016-935-8478','F','DCSY1S3');

INSERT INTO Student VALUES ('20WMD02010','LAW QIAN YEE','017-835-8478','F','DCSY1S3');

## 5.2 Branch

INSERT INTO Branch VALUES ('DCSY1S1', 'Diploma in Computer Science Year 1 Semester 1');

INSERT INTO Branch VALUES ('DCSY1S2', 'Diploma in Computer Science Year 1 Semester 2');

INSERT INTO Branch VALUES ('DCSY1S3', 'Diploma in Computer Science Year 1 Semester 3');

INSERT INTO Branch VALUES ('DCSY1S4', 'Diploma in Computer Science Year 1 Semester 4');

INSERT INTO Branch VALUES ('DCSY2S1', 'Diploma in Computer Science Year 2 Semester 1');

INSERT INTO Branch VALUES ('DCSY2S2', 'Diploma in Computer Science Year 2 Semester 2');

INSERT INTO Branch VALUES ('DCSY2S3', 'Diploma in Computer Science Year 2 Semester 3');

INSERT INTO Branch VALUES ('DCSY3S1', 'Diploma in Computer Science Year 3 Semester 1');

INSERT INTO Branch VALUES ('DCSY3S2', 'Diploma in Computer Science Year 3 Semester 2');

INSERT INTO Branch VALUES ('DCSY3S3', 'Diploma in Computer Science Year 3 Semester 3');

## 5.3 Room

INSERT INTO Room VALUES('ROOM000','STUDY ROOM 1');

INSERT INTO Room VALUES('ROOM001','STUDY ROOM 2');

INSERT INTO Room VALUES('ROOM002','MEETING ROOM 1');

INSERT INTO Room VALUES('ROOM003','MEETING ROOM 2');

INSERT INTO Room VALUES('ROOM004','COMPUTER ROOM 1');

INSERT INTO Room VALUES('ROOM005','COMPUTER ROOM 2');

INSERT INTO Room VALUES('ROOM006','CHATTING ROOM 1');

INSERT INTO Room VALUES('ROOM007','CHATTING ROOM 2');

INSERT INTO Room VALUES('ROOM008','MULTIPURPOSE ROOM 1');

INSERT INTO Room VALUES('ROOM009','MULTIPURPOSE ROOM 2');

## 5.4 Room booking details

insert into Room\_booking\_details values ('R001', 'ROOM009', '10:38', '11:41');

insert into Room\_booking\_details values ('R002', 'ROOM001', '15:41', '16:53');

insert into Room\_booking\_details values ('R003', 'ROOM005', '15:17', '16:19');

insert into Room\_booking\_details values ('R004', 'ROOM004', '08:03', '09:37');

insert into Room\_booking\_details values ('R005', 'ROOM004', '16:51', '18:23');

insert into Room\_booking\_details values ('R006', 'ROOM006', '08:34', '09:14');

insert into Room\_booking\_details values ('R007', 'ROOM009', '11:24', '12:23');

insert into Room\_booking\_details values ('R008', 'ROOM001', '16:20', '17:04');

insert into Room\_booking\_details values ('R009', 'ROOM003', '10:52', '11:52');

insert into Room\_booking\_details values ('R010', 'ROOM009', '10:43', '11:42');

## 5.5 Book Loan

INSERT INTO book\_loan values('LOID00001','01-JAN-2021','08-JAN-2021','15-JAN-2021', '20WMD02001');

INSERT INTO book\_loan values('LOID00002','01-JAN-2021','09-JAN-2021','15-JAN-2021', '20WMD02002');

INSERT INTO book\_loan values('LOID00003','01-JAN-2021','10-JAN-2021','15-JAN-2021', '20WMD02003');

INSERT INTO book\_loan values('LOID00004','02-FEB-2021','09-FEB-2021','15-FEB-2021', '20WMD02004');

INSERT INTO book\_loan values('LOID00005','02-FEB-2021','10-FEB-2021','15-FEB-2021', '20WMD02005');

INSERT INTO book\_loan values('LOID00006','02-FEB-2021','11-FEB-2021','15-FEB-2021', '20WMD02006');

INSERT INTO book\_loan values('LOID00007','04-APR-2021','11-APR-2021','18-APR-2021', '20WMD02007');

INSERT INTO book\_loan values('LOID00008','04-APR-2021','12-APR-2021','18-APR-2021', '20WMD02008');

INSERT INTO book\_loan values('LOID00009','05-MAY-2021','12-MAY-2021','19-MAY-2021', '20WMD02009');

INSERT INTO book\_loan values('LOID00010','05-MAY-2021','13-MAY-2021','19-MAY-2021', '20WMD02010');

## 5.6 Book Loan Details

INSERT INTO book\_loan\_details VALUES('LOID00001', '20WMD02001', '01-JAN-2021', '08-JAN-2021', 0.00, 'AAA0001');

INSERT INTO book\_loan\_details VALUES('LOID00002', '20WMD02002', '01-JAN-2021', '09-JAN-2021', 0.10, 'AAA0002');

INSERT INTO book\_loan\_details VALUES('LOID00003', '20WMD02003', '01-JAN-2021', '10-JAN-2021', 0.20, 'AAA0003');

INSERT INTO book\_loan\_details VALUES('LOID00004', '20WMD02004', '02-FEB-2021', '09-FEB-2021', 0.00, 'AAA0001');

INSERT INTO book\_loan\_details VALUES('LOID00005', '20WMD02005', '02-FEB-2021', '10-FEB-2021', 0.10, 'AAA0004');

INSERT INTO book\_loan\_details VALUES('LOID00006', '20WMD02006', '02-FEB-2021', '11-FEB-2021', 0.20, 'AAA0002');

INSERT INTO book\_loan\_details VALUES('LOID00007', '20WMD02007', '04-APR-2018', '11-APR-2018', 0.00, 'AAA0005');

INSERT INTO book\_loan\_details VALUES('LOID00008', '20WMD02008', '04-APR-2018', '12-APR-2018', 0.10, 'AAA0006');

INSERT INTO book\_loan\_details VALUES('LOID00009', '20WMD02009', '05-MAY-2019', '12-MAY-2019', 0.00, 'AAA0002');

INSERT INTO book\_loan\_details VALUES('LOID00010', '20WMD02010', '05-MAY-2018', '13-MAY-2018', 0.10, 'AAA0003');

## 5.7 Book

INSERT INTO book VALUES('AAA0001','Pride and Prejudice','Gram Gwatkins', 21.1, '434492654-4', 'Romance');

INSERT INTO book VALUES('AAA0002','To Kill a Mockingbird','Amalia Girdlestone', 23.09, '520310587-1','Thriller');

INSERT INTO book VALUES('AAA0003','1984','Helge Baxandall', 48.74, '941239670-8','Science');

INSERT INTO book VALUES('AAA0004','Harry Potter','Ingamar Ebbers', 72.53, '007120577-2','Fantasy');

INSERT INTO book VALUES('AAA0005','The Hobbit','Ciro Robens', 89.14, '598103545-5','Fantasy');

INSERT INTO book VALUES('AAA0006','The Lord of the Rings','Robbie Birtle', 53.34, '294341323-2','Fantasy');

INSERT INTO book VALUES('AAA0007','Wutheing Heights','Joyann Plowright', 33.24, '452161941-X','Tragedy');

INSERT INTO book VALUES('AAA0008','Jane Eyre','Massimiliano Slarke', 20.72, '919194681-6','Romance');

INSERT INTO book VALUES('AAA0009','The Lion','Jerrold Wickes', 63.88, '954363904-3','Fantasy');

INSERT INTO book VALUES('AAA0010','The Hitchhikers','Haze Crebott', 92.76, '997715233-0','Humour');

## 5.8 Staff

INSERT INTO staff VALUES('S001','Tan Hann Ren','012-3456789','M');

INSERT INTO staff VALUES('S002','Ng Zhe Xuan','012-3669854','M');

INSERT INTO staff VALUES('S003','Sing Wen Xuan','012-7894563','F');

INSERT INTO staff VALUES('S004','Chua Wei Yun','012-7744589','F');

INSERT INTO staff VALUES('S005','Lua Shi Qi','011-2589334','F');

INSERT INTO staff VALUES('S006','Lim Cheng Zhi','011-4523601','M');

INSERT INTO staff VALUES('S007','Phong Ting Wei','011-1224578','F');

INSERT INTO staff VALUES('S008','Tan Rui Shein','014-7741258','M');

INSERT INTO staff VALUES('S009','Tay Yi Sin','012-9966584','M');

INSERT INTO staff VALUES('S010','Tay Yi Chee','012-4452369','M');

## 5.9 Room booking

insert into Room\_booking values ('R001', '08-Nov-2019', '09-Nov-2019', 'S001', '20WMD02003');

insert into Room\_booking values ('R002', '21-Feb-2020', '22-Feb-2020', 'S002', '20WMD02005');

insert into Room\_booking values ('R003', '12-May-2019', '23-May-2019', 'S003', '20WMD02006');

insert into Room\_booking values ('R004', '09-Jan-2021', '22-Jan-2021', 'S004', '20WMD02008');

insert into Room\_booking values ('R005', '06-Aug-2019', '08-Aug-2019', 'S005', '20WMD02010');

insert into Room\_booking values ('R006', '05-Jan-2021', '20-Jan-2021', 'S006', '20WMD02011');

insert into Room\_booking values ('R007', '10-Jan-2020', '21-Jan-2020', 'S007', '20WMD02015');

insert into Room\_booking values ('R008', '04-Aug-2019', '16-Aug-2019', 'S008', '20WMD02016');

insert into Room\_booking values ('R009', '12-Sep-2020', '13-Sep-2020', 'S009', '20WMD02018');

insert into Room\_booking values ('R010', '10-Nov-2019', '26-Nov-2019', 'S010', '20WMD02019');

# Task 6: SQL Queries and Reports

## 6.1 Low Jun Yan

### **6.1.1 Query 1: Detail report of student using library rooms in a range of date**

Purpose: The purpose of the report is to let user search for room use history between the room use dates input and students who use the rooms between the range of date will be displayed so user can know at which time and who using the room. In addition, if someone reported their stuff left in the library room, they can refer to the system display and know who used the room after them.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY';

cl scr

PROMPT 'Data entry for Room use date'

PROMPT

PROMPT

ACCEPT v\_startDate CHAR FORMAT 'A11' PROMPT ' Enter Room Use start date: '

ACCEPT v\_endDate CHAR FORMAT 'A11' PROMPT ' Enter Room Use end date: '

COLUMN Room\_ID FORMAT A7 HEADING "ROOM ID";

COLUMN Student\_ID FORMAT A10 HEADING "STUDENT ID";

COLUMN room\_use\_date FORMAT A12 HEADING "USE DATE";

COLUMN student\_name FORMAT A20 HEADING "STUDENT NAME";

COLUMN student\_contact FORMAT A15 HEADING "STUDENT CONTACT";

TTITLE CENTER 'Room Use Details for ' \_DATE -

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON Room\_ID SKIP 2 ON Student\_ID

SELECT Room\_ID, S.Student\_ID, room\_use\_date, student\_name, student\_contact

FROM Room\_booking R, Room\_booking\_details RD, Student S

WHERE R.Room\_booking\_ID = RD.Room\_booking\_ID AND R.Student\_ID = S.Student\_ID AND room\_use\_date BETWEEN '&v\_startDate' AND '&v\_endDate'

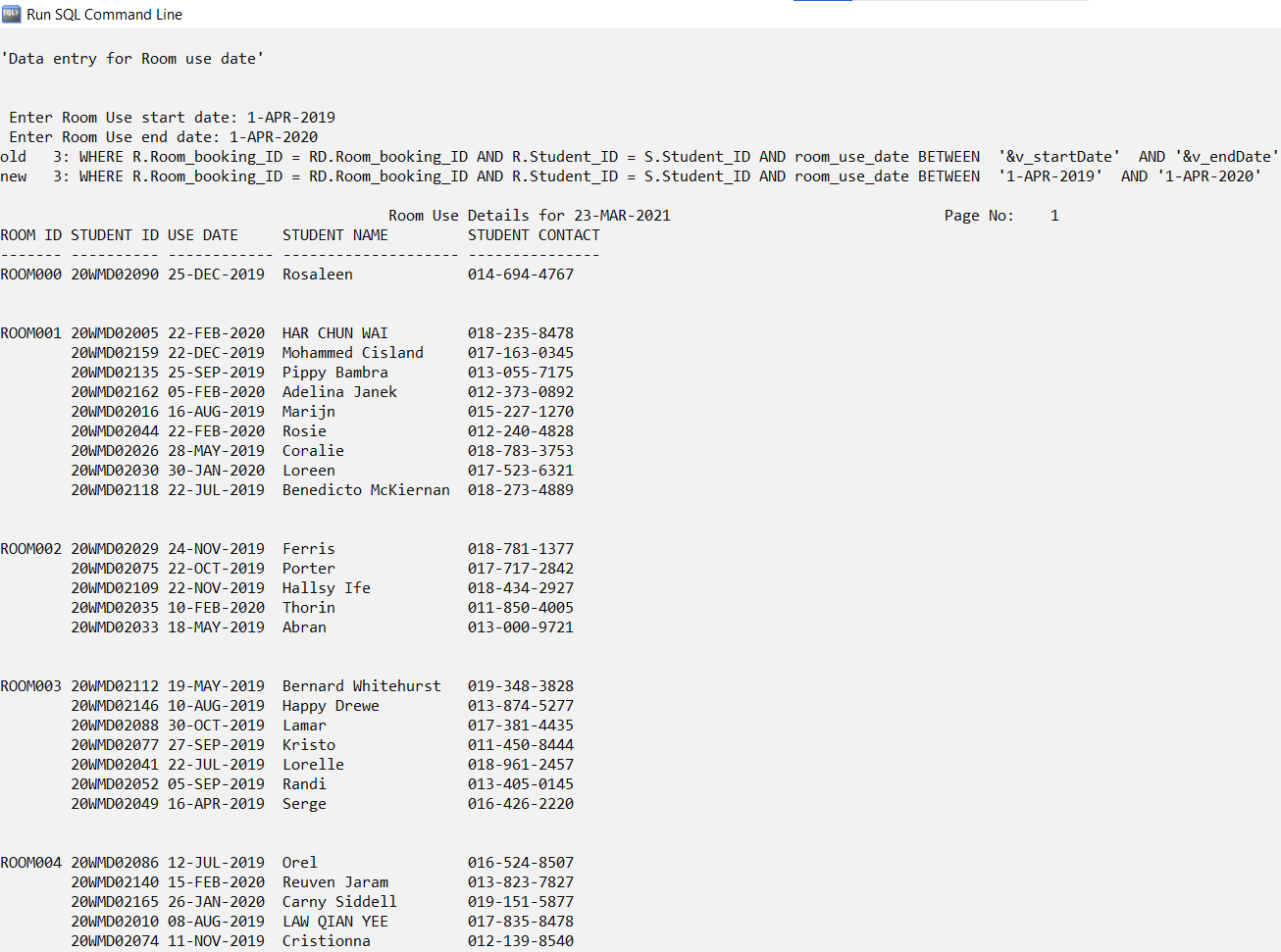
ORDER BY Room\_ID;

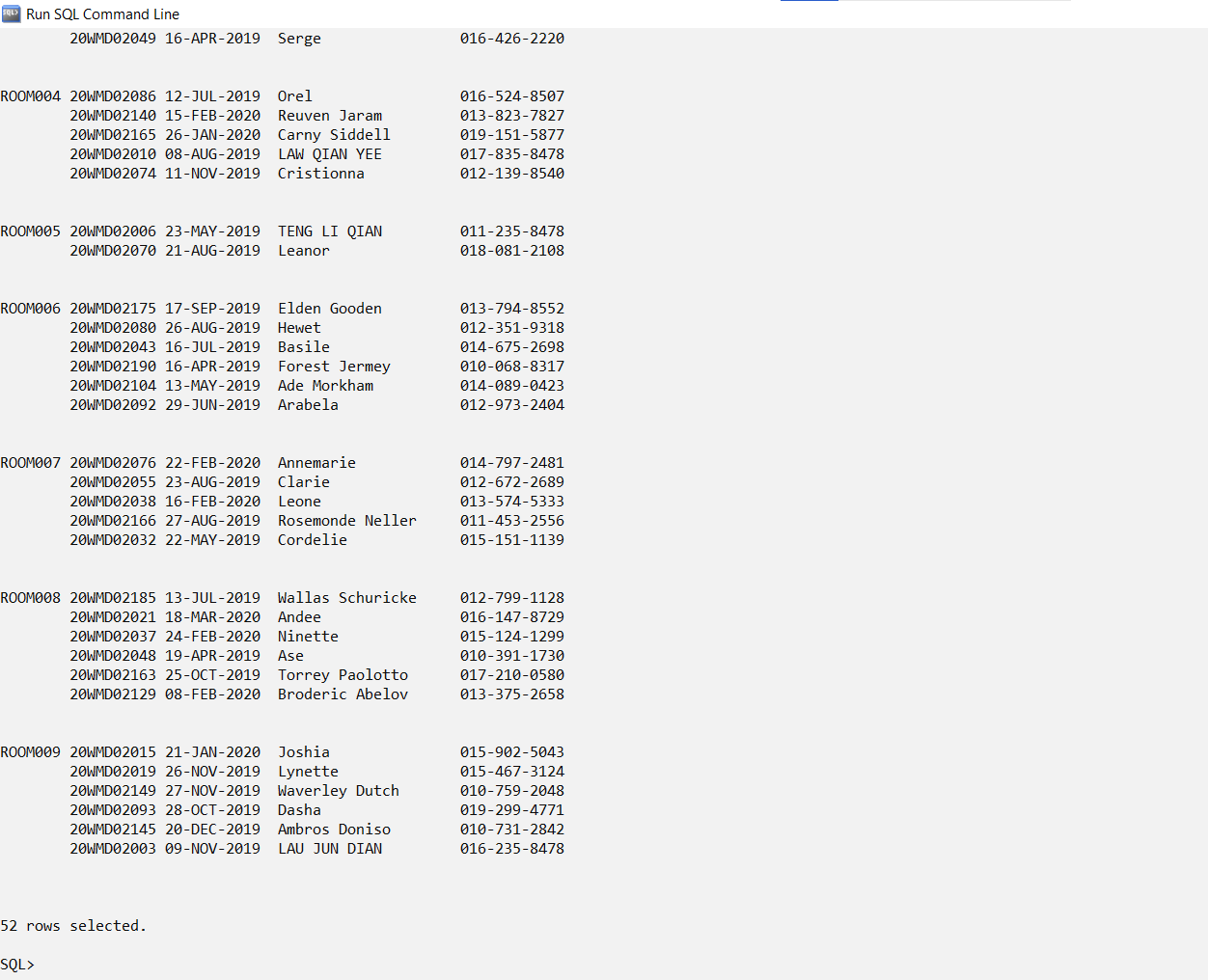
CLEAR COLUMNS

CLEAR BREAKS

TTITLE OFF

Sample Output: Screenshot





### **6.1.2 Query 2: Detail report of library rooms usage records**

Purpose: The purpose of the report is to let user to search for particular room with room ID. Records in ascending order of its use date will be displayed so user can check for student who used the room with their details (name and contact) and the date they used.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY';

cl scr

PROMPT 'Data entry for Room booking ID'

PROMPT

PROMPT

ACCEPT v\_Room\_ID CHAR FORMAT 'A7' PROMPT ' Enter Room ID(ROOM000 ~ ROOM009): '

COLUMN Room\_ID FORMAT A7 HEADING "ROOM ID";

COLUMN Student\_ID FORMAT A10 HEADING "STUDENT ID";

COLUMN room\_use\_date FORMAT A12 HEADING "USE DATE";

COLUMN student\_name FORMAT A20 HEADING "STUDENT NAME";

COLUMN student\_contact FORMAT A15 HEADING "STUDENT CONTACT";

TTITLE CENTER 'Room Booking Details for ' \_DATE -

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON Room\_ID SKIP 2 ON Student\_ID

SELECT Room\_ID, S.Student\_ID, room\_use\_date, student\_name, student\_contact

FROM Room\_booking R, Room\_booking\_details RD, Student S

WHERE R.Room\_booking\_ID = RD.Room\_booking\_ID AND R.Student\_ID = S.Student\_ID AND RD.Room\_ID = '&v\_Room\_ID'

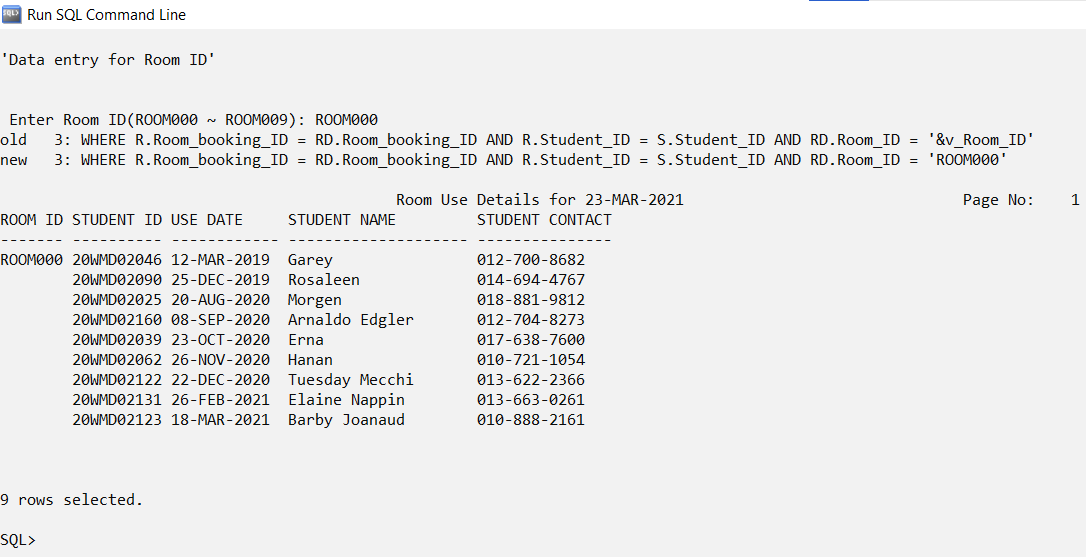
ORDER BY room\_use\_date;

CLEAR COLUMNS

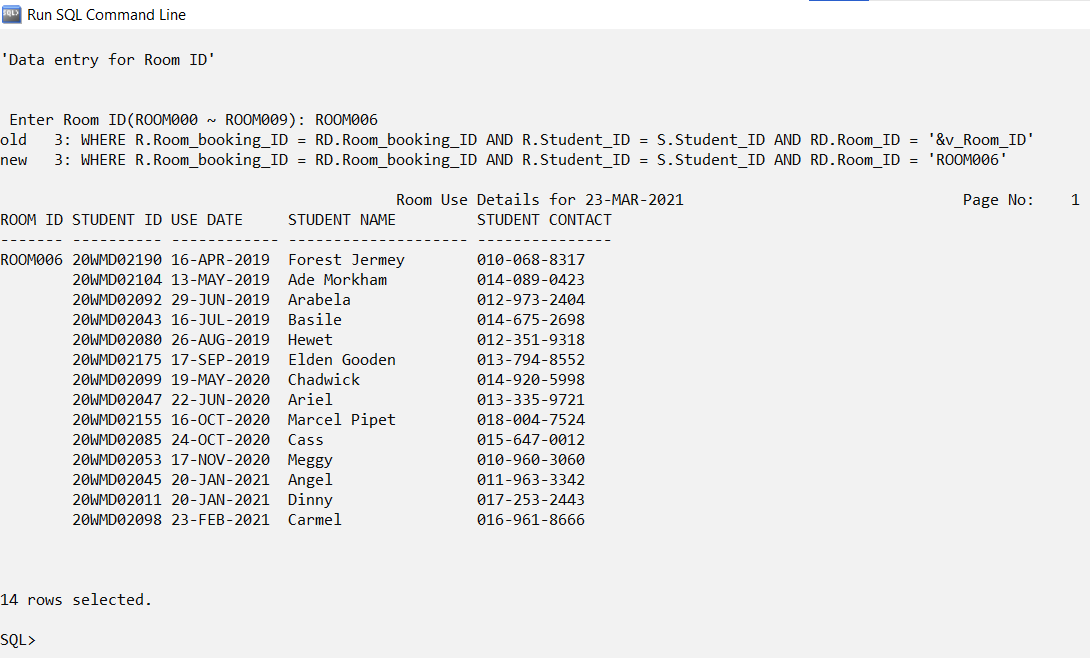
CLEAR BREAKS

TTITLE OFF

Sample Output: Screenshot 1



Sample Output: Screenshot 2



### **6.1.3 Query 3: Detail report of staff with room use date**

Purpose: The purpose of the report is to let user to have a look on which staff will be handled the libraries room on a particular date so user can contact the staff if any emergency happens. For example, when student suddenly can't use the room they booked, they can contact the staff to cancel it. In addition, the staff leader can also make better decision on assigning other task for staff who have more free time.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY';

cl scr

PROMPT 'Data entry for Staff ID'

PROMPT

PROMPT

ACCEPT v\_Staff\_ID CHAR FORMAT 'A4' PROMPT ' Enter Staff ID(S001 ~ S010): '

COLUMN Staff\_ID FORMAT A8 HEADING "STAFF ID";

COLUMN Staff\_name FORMAT A20 HEADING "STAFF NAME";

COLUMN Staff\_contact FORMAT A13 HEADING "STAFF CONTACT";

COLUMN Room\_booking\_ID FORMAT A10 HEADING "BOOKING ID";

COLUMN Room\_use\_date FORMAT A11 HEADING "USE DATE";

TTITLE CENTER 'Staff ID for ' \_DATE -

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

SELECT S.Staff\_ID, Staff\_name, Staff\_contact, Room\_booking\_ID, Room\_use\_date

FROM Staff S, Room\_booking R

WHERE R.Staff\_ID = S.Staff\_ID AND R.Staff\_ID = '&v\_Staff\_ID'

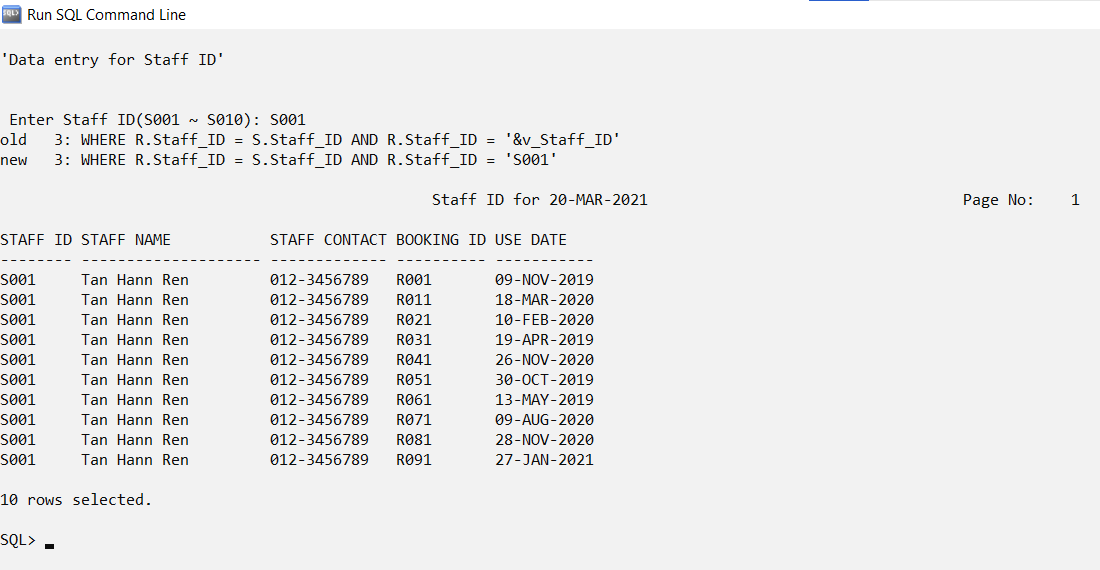
ORDER BY Room\_booking\_ID;

CLEAR COLUMNS

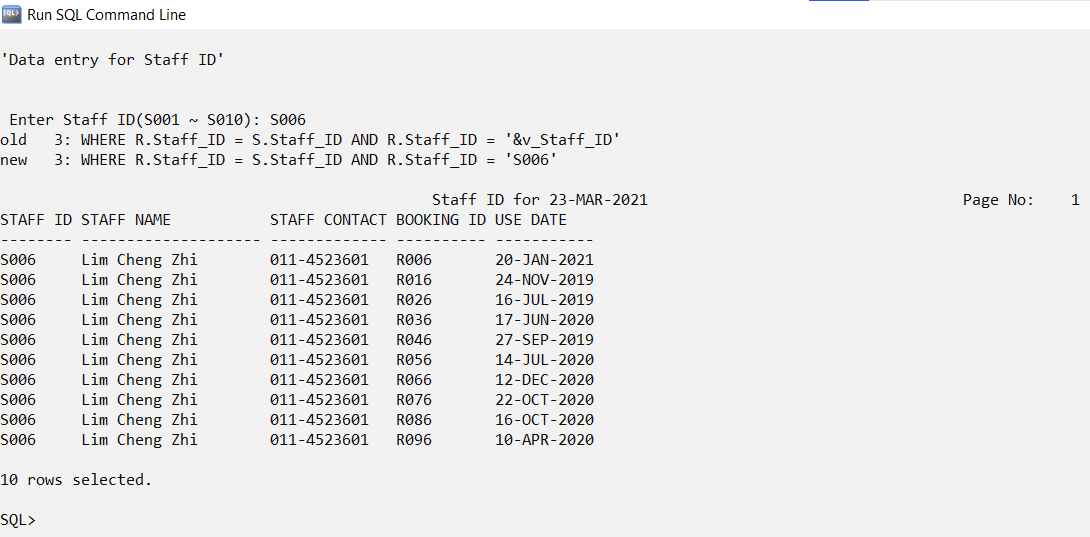
CLEAR BREAKS

TTITLE OFF

Sample Output: Screenshot 1



Sample Output: Screenshot 2



## 6.2 Tan Kang Hong

### **6.2.1 Query 1: Detail report of book with book loan id**

PURPOSE: Let user to input the book loan ID to display the records of the book loan to be viewed.

IMPORTANCE: User can search for book loan ID so user can check which book and what times was the book has been borrowed by students.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MM-YYYY';

cl scr

PROMPT 'Data entry for Book Loan ID'

PROMPT

PROMPT

ACCEPT v\_BOOK\_LOAN\_ID char FORMAT 'A14' PROMPT 'Enter Book Loan ID : '

COLUMN book\_loan\_id FORMAT A14 HEADING 'Book Loan ID'

COLUMN borrowed\_date FORMAT A14 HEADING 'Date Of Borrow'

COLUMN returned\_date FORMAT A14 HEADING 'Date Of Return'

COLUMN date\_due\_for\_return FORMAT A18 HEADING 'Date Due Of Return'

COLUMN student\_id FORMAT 'A14' HEADING 'Student ID'

TTITLE CENTER 'BOOK LOAN FOR '\_DATE-

RIGHT'Page:'FORMAT 999 SQL.PNO SKIP 2

BREAK ON book\_loan\_id SKIP 1 ON student\_id

SELECT B.book\_loan\_id,borrowed\_date,returned\_date,date\_due\_for\_return,S.student\_id

FROM book\_loan B,student S

WHERE B.student\_id = S.student\_id AND book\_loan\_id = '&v\_BOOK\_LOAN\_ID'

ORDER BY book\_loan\_id;

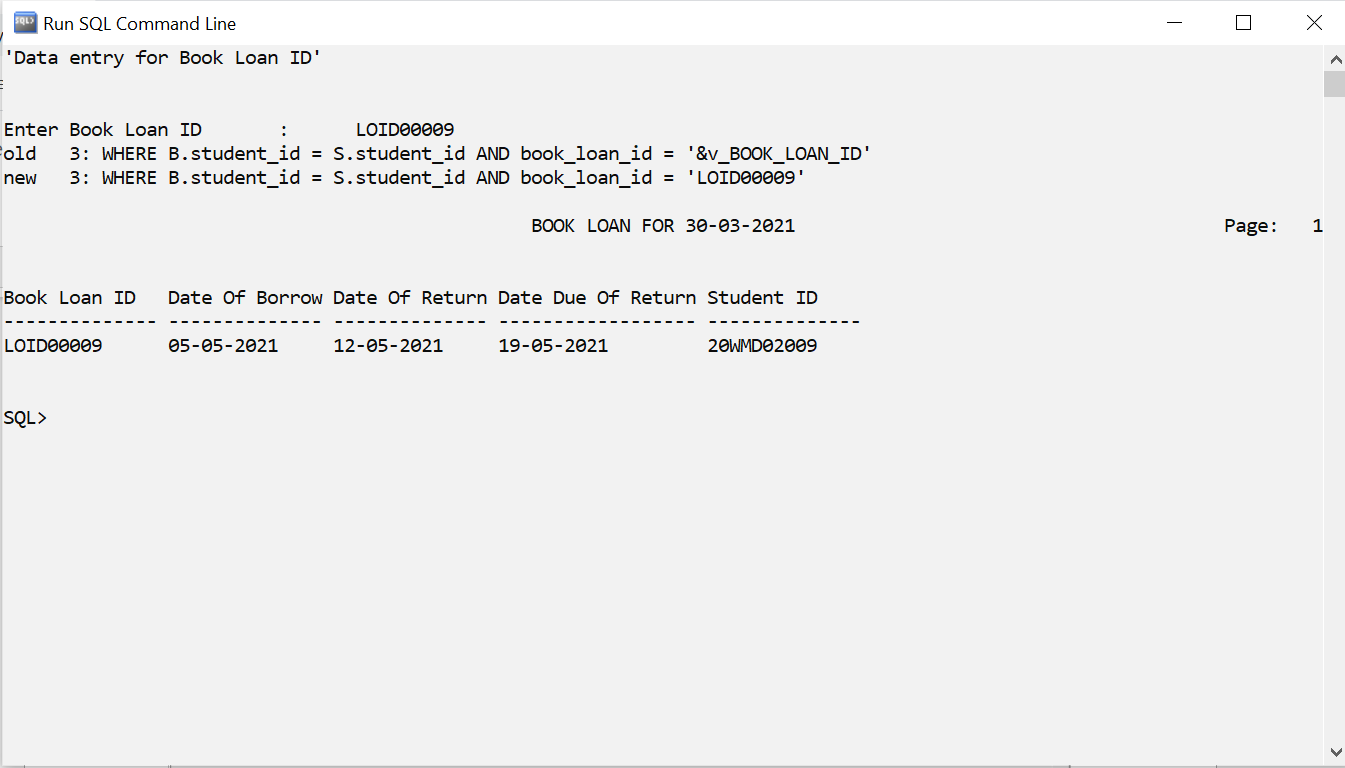
CLEAR COLUMNS

CLEAR BREAKS

CLEAR COMPUTES

TTITLE OFF

Sample Output: Screenshot



### **6.2.2 Query 2: Detail report of book with start and end borrowed date**

PURPOSE: Let user to input the borrowed start date and borrow end date to display the records between the dates input.

IMPORTANCE: User can search for the borrowed date and in the adjustable range with ascending order of book loan ID so user can check totally how many students borrow book during this period.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MM-YYYY';

cl scr

PROMPT 'Data entry for Borrowed Date'

PROMPT

PROMPT

ACCEPT v\_START\_BORROWED\_DATE char FORMAT 'A14' PROMPT 'Enter Start Borrowed Date : '

ACCEPT v\_END\_BORROWED\_DATE char FORMAT 'A14' PROMPT 'Enter End Borrowed Date : '

COLUMN book\_loan\_id FORMAT A14 HEADING 'Book Loan ID'

COLUMN borrowed\_date FORMAT A14 HEADING 'Date Of Borrow'

COLUMN returned\_date FORMAT A14 HEADING 'Date Of Return'

COLUMN date\_due\_for\_return FORMAT A18 HEADING 'Date Due Of Return'

COLUMN student\_id FORMAT 'A14' HEADING 'Student ID'

TTITLE CENTER 'BOOK LOAN FOR '\_DATE-

RIGHT'Page:'FORMAT 999 SQL.PNO SKIP 2

BREAK ON book\_loan\_id SKIP 1 ON student\_id

SELECT B.book\_loan\_id,borrowed\_date,returned\_date,date\_due\_for\_return,S.student\_id

FROM book\_loan B,student S

WHERE B.student\_id = S.student\_id AND borrowed\_date BETWEEN '&v\_START\_BORROWED\_DATE' AND '&v\_END\_BORROWED\_DATE' AND ROWNUM <= 10

ORDER BY book\_loan\_id;

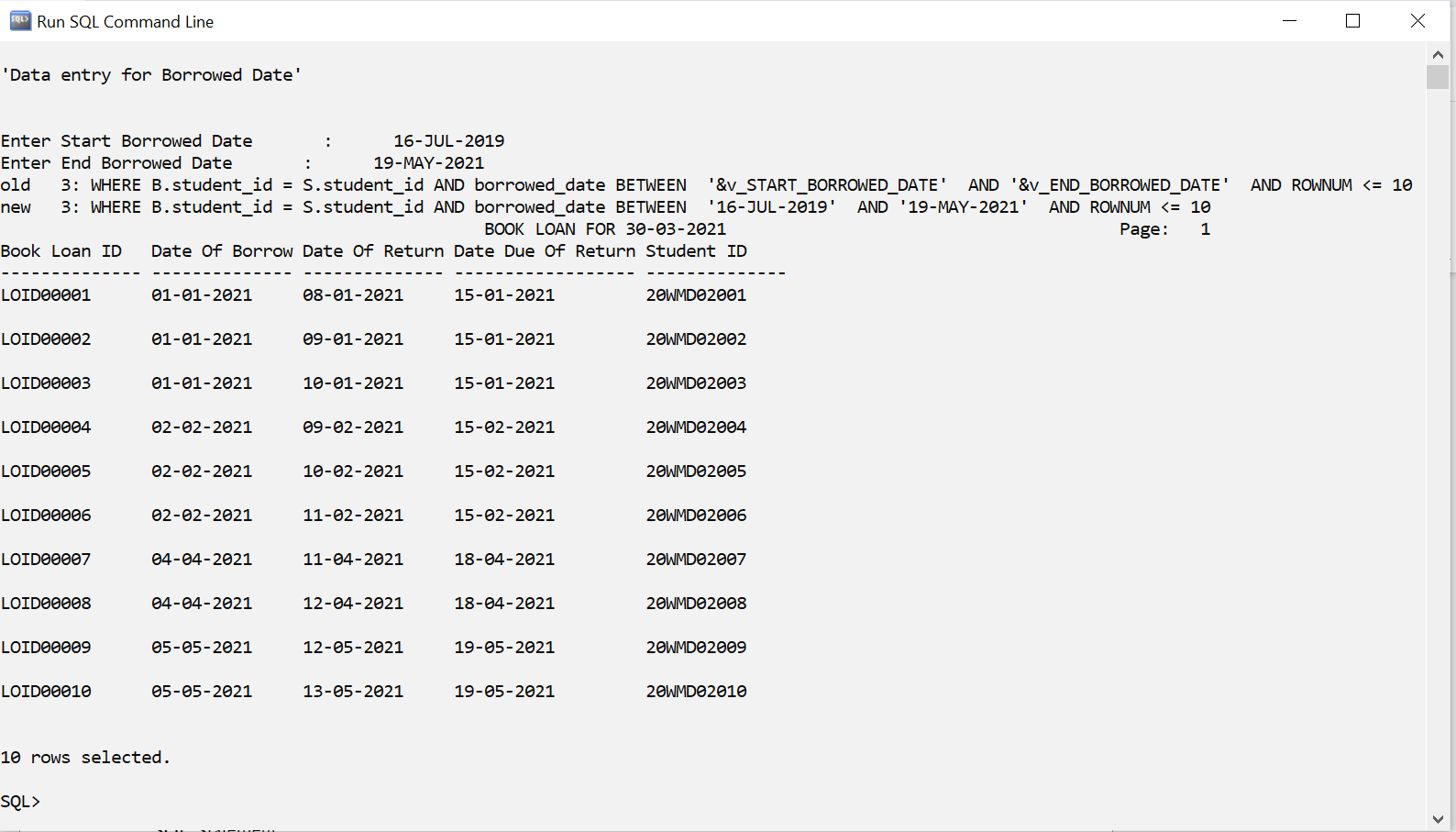
CLEAR COLUMNS

CLEAR BREAKS

CLEAR COMPUTES

TTITLE OFF

Sample Output: Screenshot



### **6.2.3 Query 3: Detail report of book with search start and end due date**

PURPOSE: Let user to input the due for return start date and due for return end date to display the records between the dates input.

IMPORTANCE: User can search for the due date for borrowed and in the adjustable range with ascending order of book loan ID so user can easy to check how many student return books before due date.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MM-YYYY';

cl scr

PROMPT 'Data entry for Due Date For Return'

PROMPT

PROMPT

ACCEPT v\_START\_DUE\_DATE\_FOR\_RETURN char FORMAT 'A18' PROMPT 'Enter Start Due Date For Return : '

ACCEPT v\_END\_DUE\_DATE\_FOR\_RETURN char FORMAT 'A18' PROMPT 'Enter End Due Date For Return : '

COLUMN book\_loan\_id FORMAT A14 HEADING 'Book Loan ID'

COLUMN borrowed\_date FORMAT A14 HEADING 'Date Of Borrow'

COLUMN returned\_date FORMAT A14 HEADING 'Date Of Return'

COLUMN date\_due\_for\_return FORMAT A18 HEADING 'Date Due Of Return'

COLUMN student\_id FORMAT 'A14' HEADING 'Student ID'

TTITLE CENTER 'BOOK LOAN FOR '\_DATE-

RIGHT'Page:'FORMAT 999 SQL.PNO SKIP 2

BREAK ON book\_loan\_id SKIP 1 ON student\_id

SELECT B.book\_loan\_id,borrowed\_date,returned\_date,date\_due\_for\_return,S.student\_id

FROM book\_loan B,student S

WHERE B.student\_id = S.student\_id AND date\_due\_for\_return BETWEEN '&v\_START\_DUE\_DATE\_FOR\_RETURN' AND '&v\_END\_DUE\_DATE\_FOR\_RETURN' AND ROWNUM <= 30

ORDER BY book\_loan\_id;

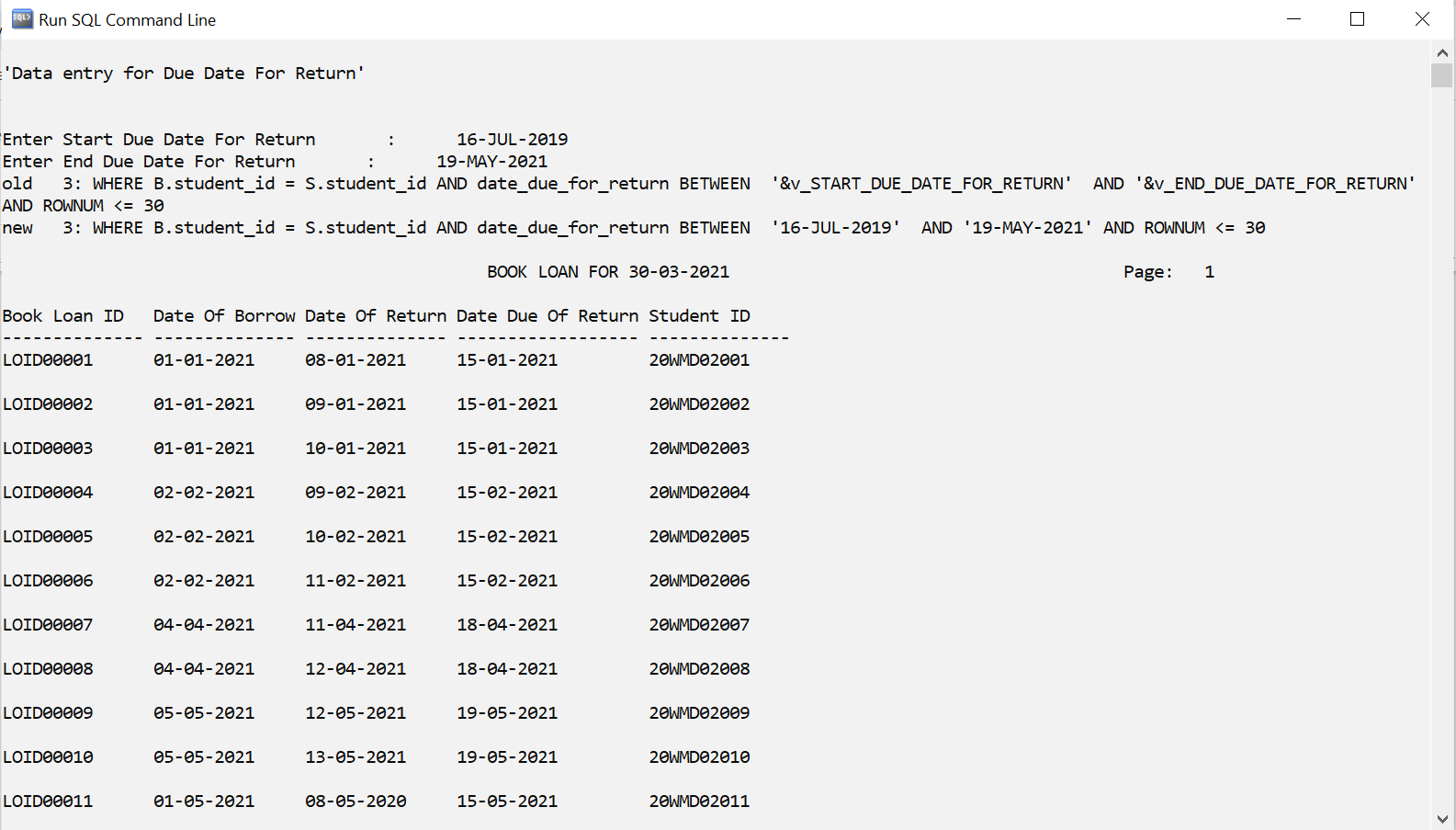
CLEAR COLUMNS

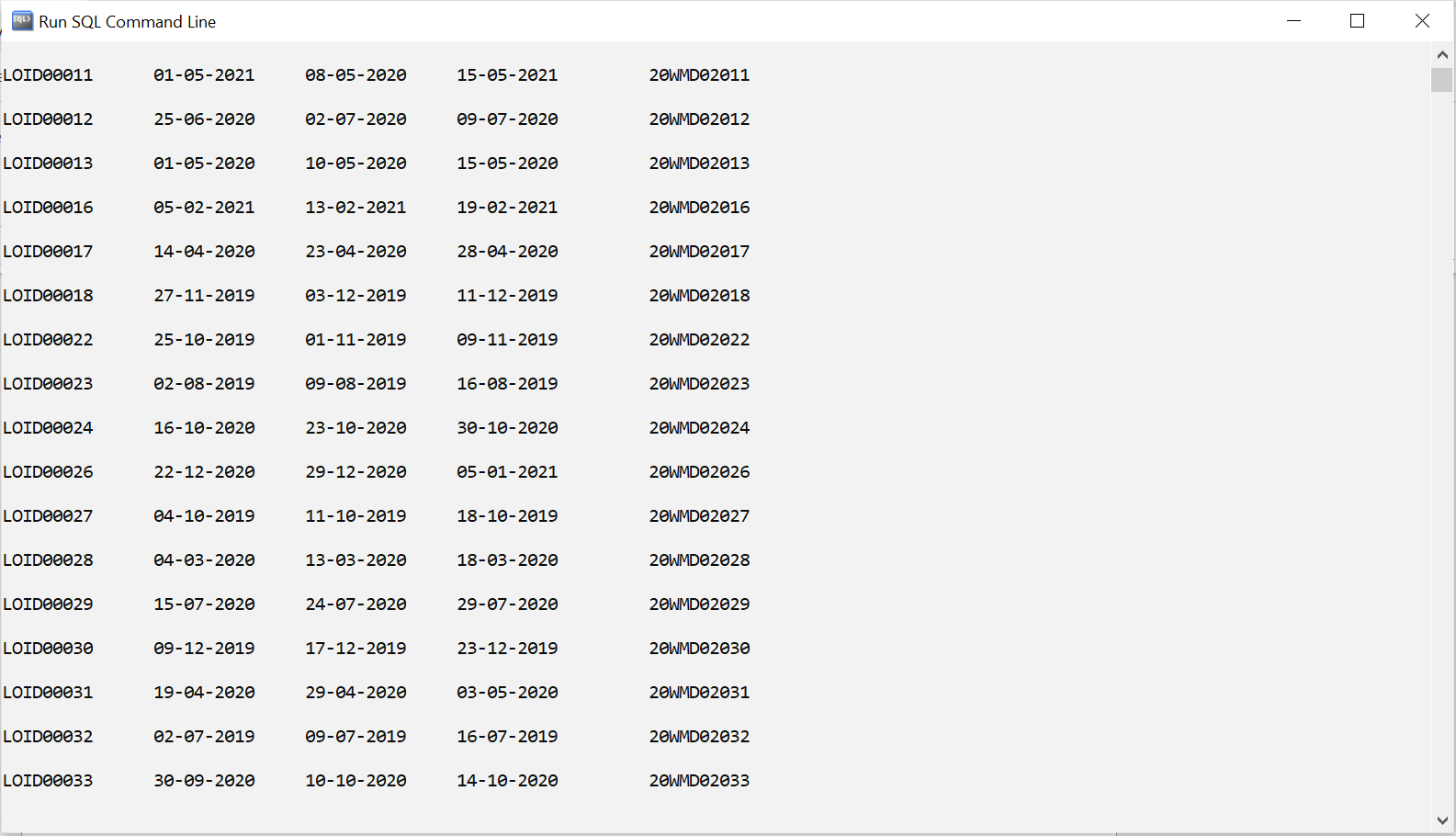
CLEAR BREAKS

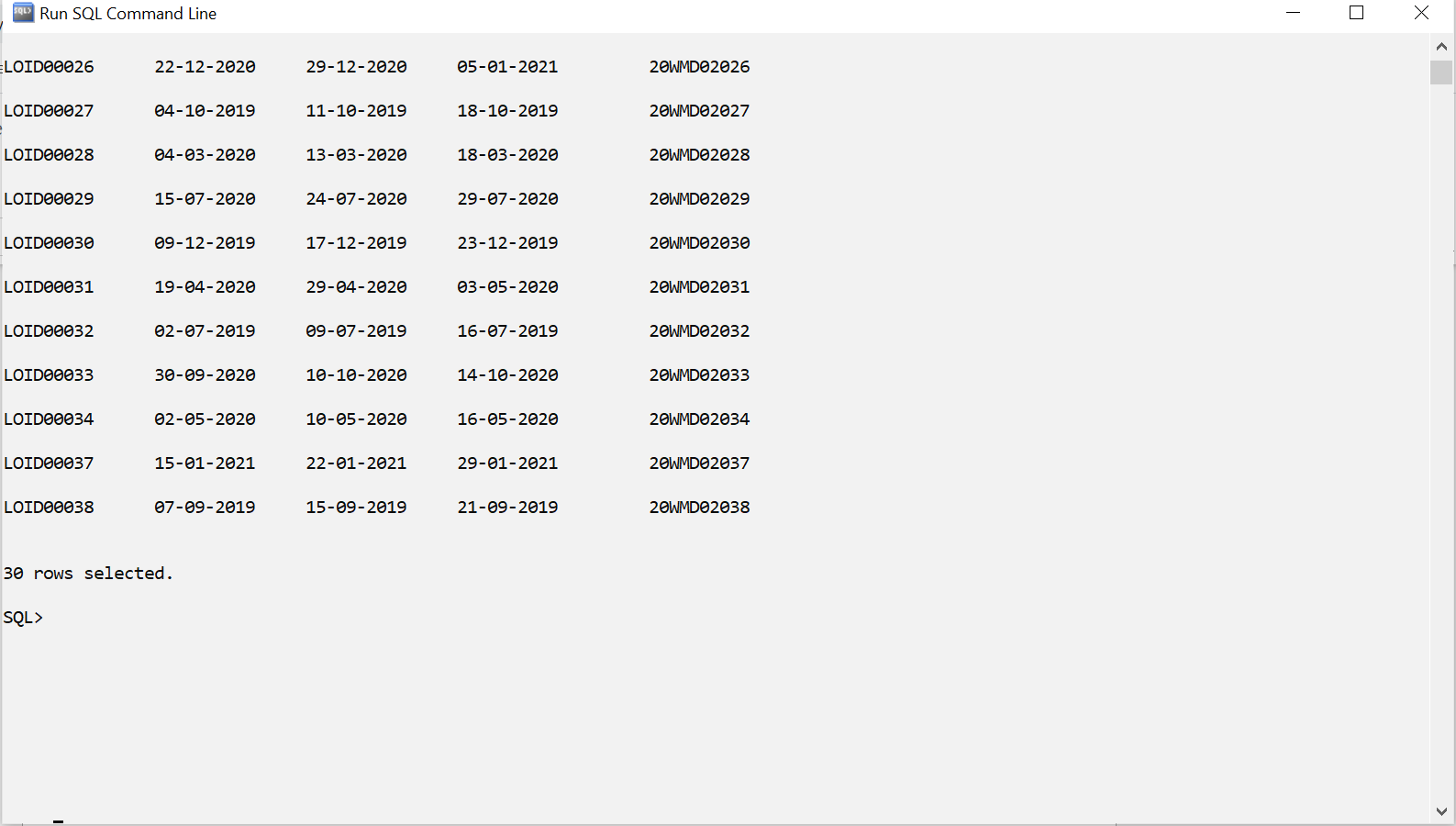
CLEAR COMPUTES

TTITLE OFF

Sample Output: Screenshot







## 6.3 Har Chun Wai

### **6.3.1 Query 1: Detail report of book with borrow and return dates**

PURPOSE: To let the user input the book ID to display the records of the book loan details to be viewed, while ordered by book loan ID.

IMPORTANCE: User can search for book ID with ascending order of book loan ID so user can check how many times that the book has been borrowed by students.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'dd-mm-yyyy';

cl scr;

PROMPT +---------------------------+

PROMPT | Select records by Book ID |

PROMPT | E.g.: AAA0010 |

PROMPT +---------------------------+

ACCEPT v\_select\_book\_id CHAR FORMAT 'A7' PROMPT ' Enter the Book ID: '

COLUMN book\_loan\_id FORMAT A12 HEADING "Book Loan ID";

COLUMN book\_id FORMAT A7 HEADING "Book ID";

COLUMN book\_title FORMAT A30 HEADING "Book Title";

COLUMN borrowed\_date FORMAT A11 HEADING "Borrow Date";

COLUMN return\_date FORMAT A11 HEADING "Return Date";

TTITLE CENTER 'Book Loan Details List' -

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON book\_id SKIP 2

SELECT BD.book\_loan\_id, BD.book\_id, B.book\_title, BD.borrowed\_date, BD.return\_date

FROM Book\_loan\_details BD, Book B, Book\_Loan BL

WHERE BD.book\_loan\_id = BL.book\_loan\_id AND BD.book\_id = B.book\_id AND BD.book\_id = '&v\_select\_book\_id'

ORDER BY BD.book\_loan\_id;

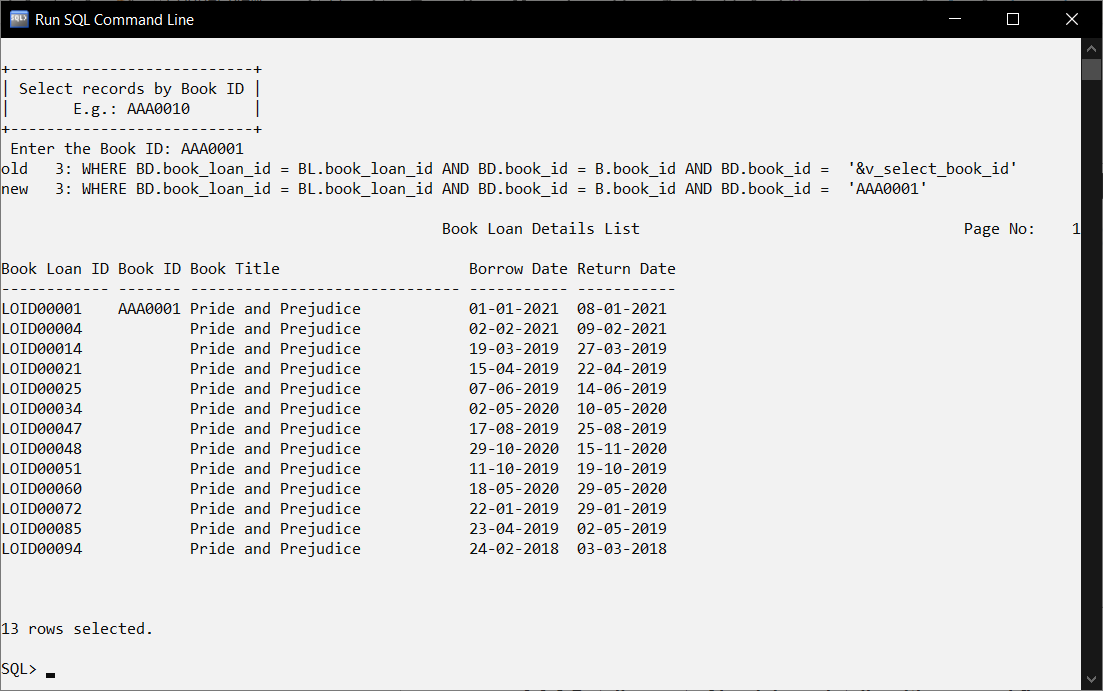
CLEAR COLUMNS

CLEAR BREAKS

CLEAR COMPUTES

TTITLE OFF

Sample Output: Screenshot



### **6.3.2 Query 2: Detail report of book loan details with grouped fine amount**

2. PURPOSE: To let the user input the minimum and maximum fine amount to display the records between the fine amount input, while ordered by book loan ID.

IMPORTANCE: User can search for the fine amount in the adjustable range with ascending order of book loan ID so user can check for students who have been fined with the amount that is within the user input fine amount.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'dd-mm-yyyy';

cl scr;

PROMPT +-------------------------------+

PROMPT | Select records by fine amount |

PROMPT | E.g.: 0.10 - 0.20 |

PROMPT +-------------------------------+

ACCEPT v\_start\_fine\_amt NUMBER FORMAT '99.99' PROMPT ' Enter the minimum fine amount: '

ACCEPT v\_end\_fine\_amt NUMBER FORMAT '99.99' PROMPT ' Enter the maximum fine amount: '

COLUMN book\_loan\_id FORMAT A12 HEADING "Book Loan ID";

COLUMN student\_id FORMAT A10 HEADING "Student ID";

COLUMN book\_id FORMAT A10 HEADING "Book ID";

COLUMN borrowed\_date FORMAT A11 HEADING "Borrow Date";

COLUMN return\_date FORMAT A11 HEADING "Return Date";

COLUMN fine FORMAT 90.99 HEADING "Fine amount (RM)";

TTITLE CENTER 'Book Loan Details List' -

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON fine SKIP PAGE

SELECT BD.book\_loan\_id, BD.student\_id, BD.book\_id, BD.borrowed\_date, BD.return\_date, BD.fine

FROM Student S, Book\_loan\_details BD, Book\_Loan BL, Book B

WHERE BD.book\_loan\_id = BL.book\_loan\_id AND BD.student\_id = S.student\_ID AND BD.book\_id = B.book\_id AND BD.fine BETWEEN '&v\_start\_fine\_amt' AND '&v\_end\_fine\_amt'

ORDER BY BD.fine;

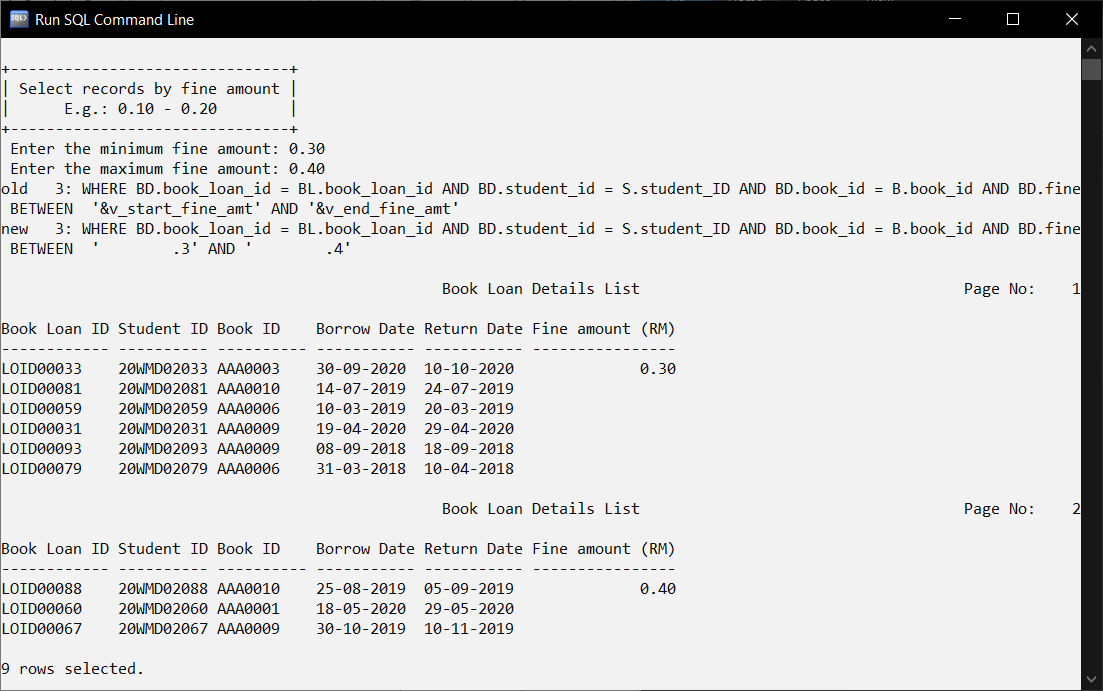
CLEAR COLUMNS

CLEAR BREAKS

CLEAR COMPUTES

TTITLE OFF

Sample Output: Screenshot



### **6.3.3 Query 3: Detail report of books borrowed by a student**

3. PURPOSE: To let the user input the student ID to display the books borrowed by the student.

IMPORTANCE: User can search for the student ID so user can check what book have been borrowed by the student with their book title, borrow and return date.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'dd-mm-yyyy';

cl scr;

PROMPT +------------------------------+

PROMPT | Select records by Student ID |

PROMPT | E.g.: 20WMD02004 |

PROMPT +------------------------------+

ACCEPT v\_select\_student\_id CHAR FORMAT 'A10' PROMPT ' Enter the student ID: '

COLUMN student\_id FORMAT A10 HEADING "Student ID";

COLUMN book\_loan\_id FORMAT A12 HEADING "Book Loan ID";

COLUMN book\_id FORMAT A10 HEADING "Book ID";

COLUMN book\_title FORMAT A30 HEADING "Book Title";

COLUMN borrowed\_date FORMAT A11 HEADING "Borrow Date";

COLUMN return\_date FORMAT A11 HEADING "Return Date";

TTITLE CENTER 'Book Loan Details List' -

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON student\_id

SELECT BD.student\_id, BD.book\_loan\_id, BD.book\_id, B.book\_title, BD.borrowed\_date, BD.return\_date

FROM Book\_loan\_details BD, Book\_Loan BL, Book B, Student S

WHERE BD.student\_id = S.student\_id AND BD.book\_loan\_id = BL.book\_loan\_id AND BD.book\_id = B.book\_id AND BD.student\_id = '&v\_select\_student\_id' AND BD.borrowed\_date = BL.borrowed\_date

ORDER BY BD.book\_loan\_id;

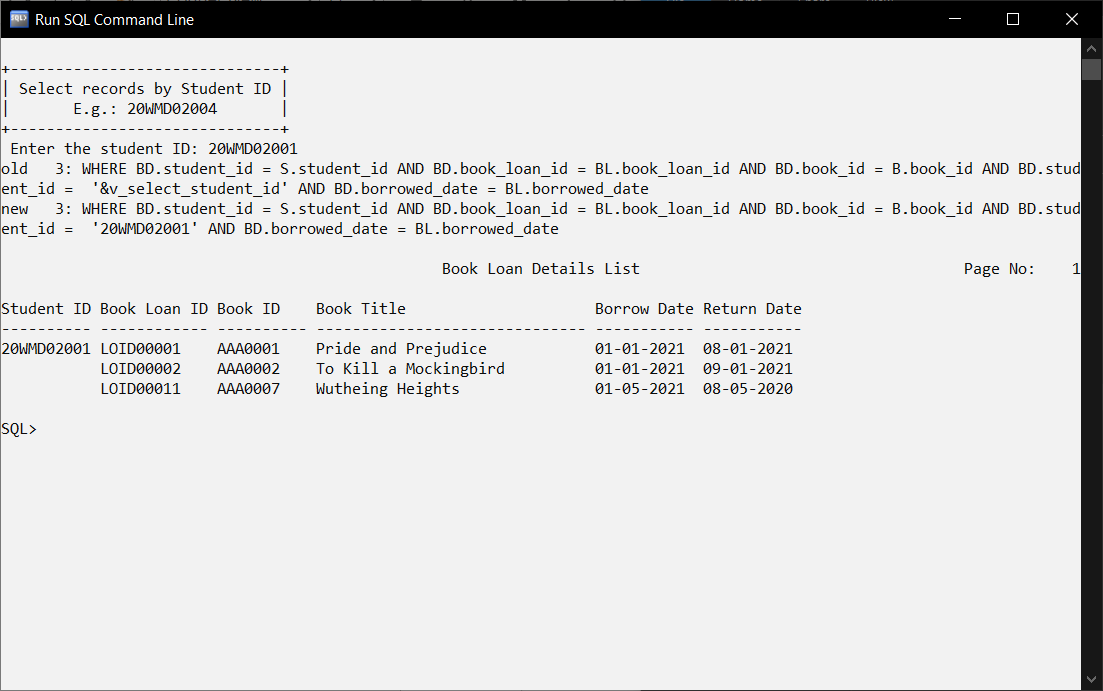
CLEAR COLUMNS

CLEAR BREAKS

CLEAR COMPUTES

TTITLE OFF

Sample Output: Screenshot 1



## 6.4 Ho Jing Xian

### **6.4.1 Query 1: Detail report of books with book borrowed date**

PURPOSE: The purpose of the report is to allow user to search for books with book borrowed date and book borrowed on that date will be displayed so user can see the records that the number of books borrow by students on that day.

IMPORTANCE: Staff can search for book borrowed date and students who borrow the book at the particular date will be displayed so staff can know who borrowed that book at that date.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY';

cl scr

PROMPT 'Data entry for Book Borrowed Date'

PROMPT

PROMPT

ACCEPT v\_startDate CHAR FORMAT 'A11' PROMPT ' Enter Book Borrowed Date: '

COLUMN book\_id FORMAT A7 HEADING "Book ID";

COLUMN student\_id FORMAT A11 HEADING "Student ID";

COLUMN student\_name FORMAT A17 HEADING "Student Name";

COLUMN book\_title FORMAT A27 HEADING "Book Title";

COLUMN borrowed\_date FORMAT A14 HEADING "Borrowed Date";

TTITLE CENTER 'Book Borrowed List for ' \_DATE -

RIGHT ' Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON book\_id SKIP 1 ON student\_id

SELECT B.book\_id, book\_title, S.student\_id, student\_name, borrowed\_date

FROM student S, book B, book\_loan\_details BD

WHERE B.book\_id = BD.book\_id AND S.student\_id = BD.student\_id AND borrowed\_date = '&v\_startDate'

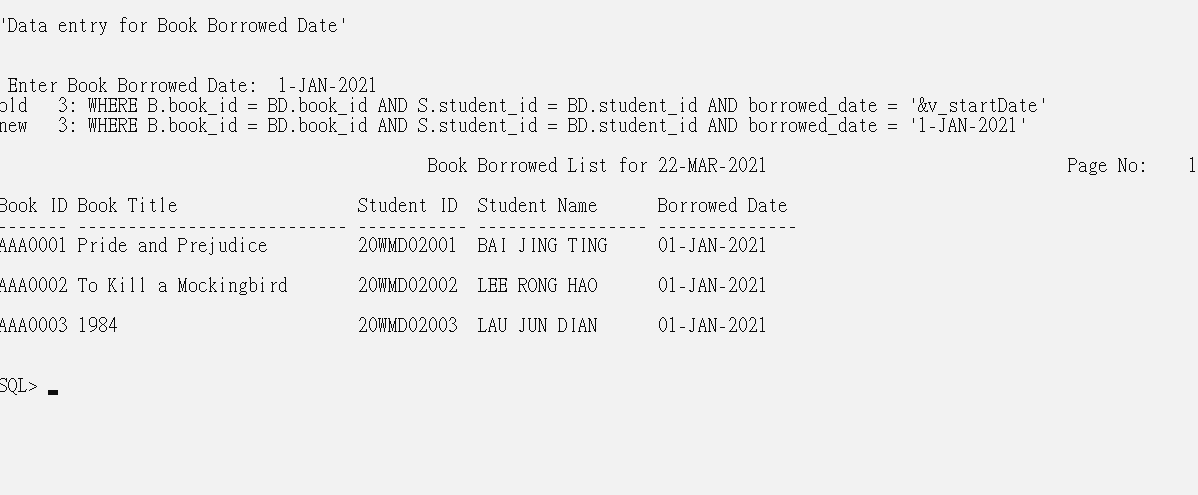
ORDER BY book\_id;

CLEAR COLUMNS

CLEAR BREAK

TTITLE OFF

Sample Output: Screenshot



### **6.4.2 Query 2: Detail report of student with book categories**

PURPOSE: The purpose of the report is to allow user to search the book with book categories. The record of book id and student id will be displayed in ascending order so that user can easily check the records that the number of students borrowed book in that categories.

IMPORTANCE: Staff can search for the book categories so staff can check these categories of book have been borrowed by the which student with the book title, student name and student id.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY';

cl scr

PROMPT 'Data entry for Categories'

PROMPT

PROMPT

ACCEPT b\_categories CHAR FORMAT 'A15' PROMPT ' Enter Categories : '

COLUMN book\_id FORMAT A7 HEADING "Book ID";

COLUMN book\_title FORMAT A28 HEADING "Book Title";

COLUMN student\_id FORMAT A10 HEADING "Student ID";

COLUMN student\_name FORMAT A20 HEADING "Student Name";

COLUMN categories FORMAT A15 HEADING "Categories";

TTITLE CENTER 'Book List for ' \_DATE -

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON book\_id SKIP 1 ON student\_id

SELECT B.book\_id, book\_title, S.student\_id, student\_name, categories

FROM student S, book B, book\_loan\_details BD

WHERE B.book\_id = BD.book\_id AND BD.student\_id = S.student\_id AND categories = '&b\_categories'

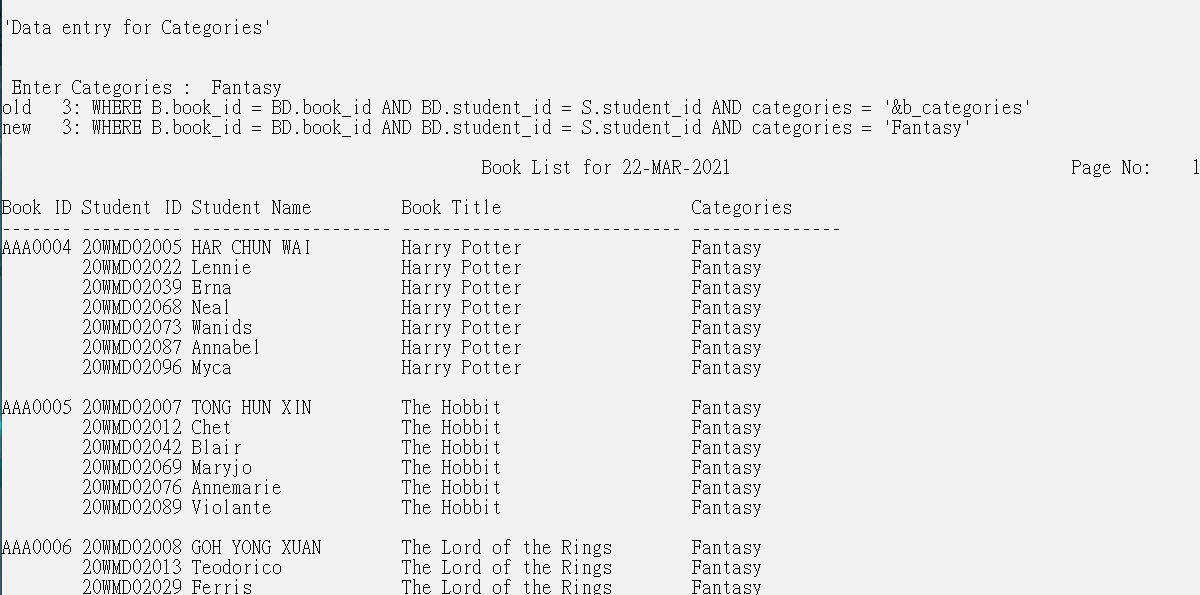
ORDER BY book\_id, student\_id;

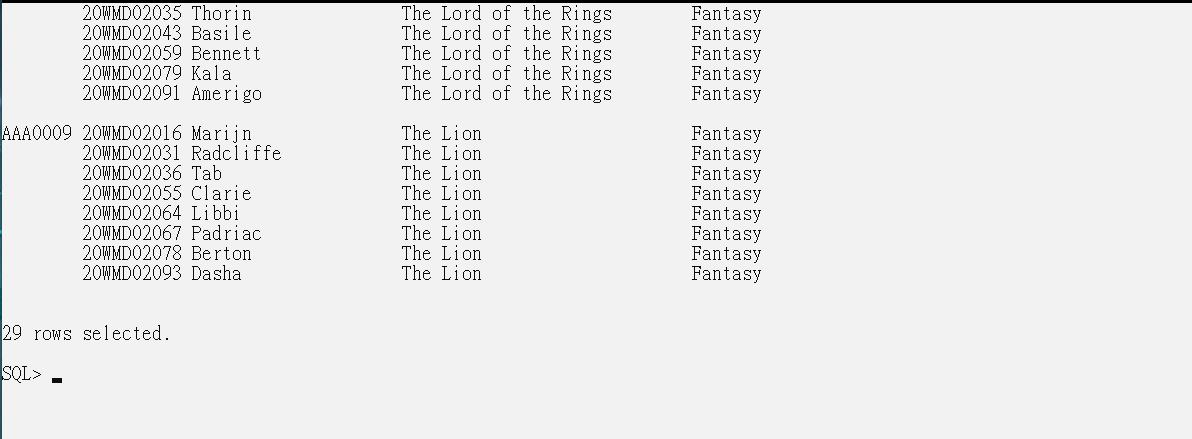
CLEAR COLUMNS

CLEAR BREAK

TTITLE OFF

Sample Output: Screenshot





### **6.4.3 Query 3: Detail report of book borrowed within the range of student ID**

PURPOSE: The purpose of the report is to allow user to search the book borrowed by student within the range of student id and student in this range will be displayed so user can see the number of books borrowed between these students.

IMPORTANCE: User can search for student id in the range and records with ascending order of book id will be displayed so user can check for the number of books borrow between these students.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY';

cl scr

PROMPT 'Data entry for Student ID'

PROMPT

PROMPT

ACCEPT fstudent\_ID CHAR FORMAT 'A11' PROMPT ' Enter First Student ID: '

ACCEPT lstudent\_ID CHAR FORMAT 'A11' PROMPT ' Enter Last Student ID: '

COLUMN student\_id FORMAT A10 HEADING "Student ID";

COLUMN student\_name FORMAT A20 HEADING "Student Name";

COLUMN student\_contact FORMAT A15 HEADING "Student Contact";

COLUMN book\_id FORMAT A7 HEADING "Book ID";

COLUMN book\_title FORMAT A30 HEADING "Book Title";

TTITLE CENTER 'Student Borrow Book List for ' \_DATE -

RIGHT ' Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON book\_id SKIP 1 ON student\_id

SELECT S.student\_id, student\_name, student\_contact, B.book\_id, book\_title

FROM student S, book B, book\_loan\_details BD

WHERE B.book\_id = BD.book\_id AND BD.student\_id = S.student\_id AND S.student\_id BETWEEN '&fstudent\_ID' AND '&lstudent\_ID'

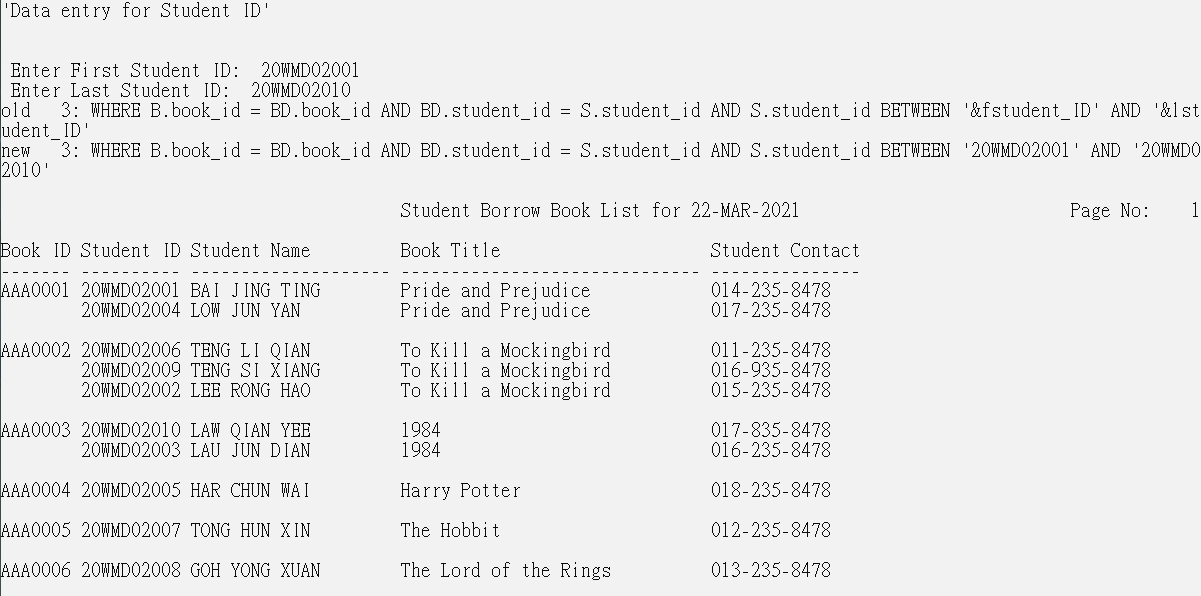
ORDER BY book\_id;

CLEAR COLUMNS

CLEAR BREAK

TTITLE OFF

Sample Output: Screenshot



## 6.5 Lau Jun Dian

### **6.5.1 Query 1: Detail report of the branch student booking which room**

PURPOSE: Let users enter the branch ID to display room student reservations.

IMPORTANCE: To know what rooms the students of which branch have booking.

SQL statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY';

ACCEPT B\_ID char FORMAT 'A8' PROMPT 'Enter Branch ID : '

COLUMN Student\_ID FORMAT A10 HEADING "Student ID ";

COLUMN Student\_name FORMAT A15 HEADING "Student Name";

COLUMN Student\_contact FORMAT A14 HEADING "Student contact";

COLUMN Student\_gender FORMAT A6 HEADING "Gender";

COLUMN Branch\_ID FORMAT A9 HEADING "Branch ID";

COLUMN Room\_booking\_ID FORMAT A15 HEADING "Room booking ID";

TTITLE CENTER 'Student &B\_ID booking room list '\_DATE -

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON Branch\_ID ON Room\_ID ON Room\_name SKIP 1 ON Student\_ID

SELECT S.Branch\_ID,R.Room\_ID,room\_name,RB.Room\_booking\_ID,S.Student\_ID,Student\_name,Student\_contact,Student\_gender

FROM Student S,Branch B,Room\_booking RB,Room\_booking\_details RBD,Room R

WHERE S.Branch\_ID=B.Branch\_ID AND RB.Student\_ID=S.Student\_ID AND RB.Room\_booking\_ID=RBD.Room\_booking\_ID AND R.Room\_ID=RBD.Room\_ID AND B.Branch\_ID='&B\_ID'

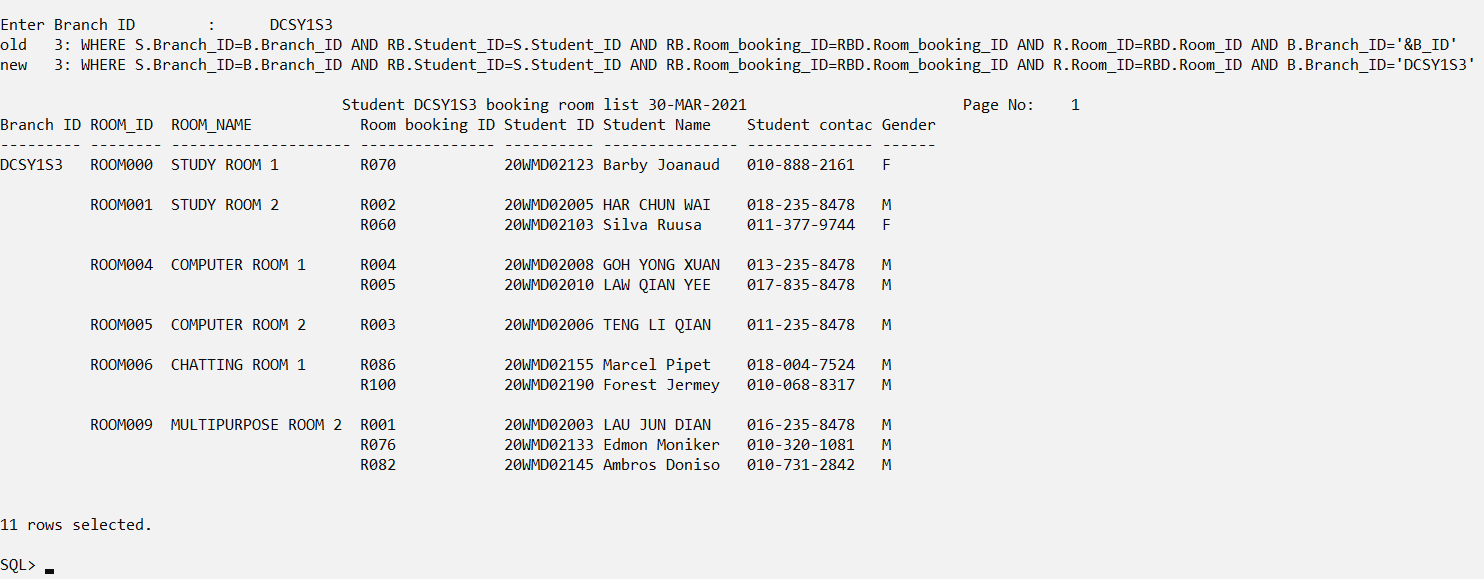
ORDER BY Room\_ID;

CLEAR COLUMNS

CLEAR BREAKS

TTITLE OFF

Sample Output: Screenshot



### **6.5.2 Query 2: Detail report of staff handle rooms booking for branch students**

PURPOSE: Let the user enter the branch ID to display the staff responsible for booking the room for the student.

IMPORTANCE: In order to know that the student who reserved the room was handled by which staff.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY';

ACCEPT B\_ID char FORMAT 'A8' PROMPT 'Enter Branch ID : '

COLUMN Student\_ID FORMAT A10 HEADING "Student ID ";

COLUMN Student\_name FORMAT A15 HEADING "Student Name";

COLUMN Student\_contact FORMAT A14 HEADING "Student Contact";

COLUMN Student\_gender FORMAT A6 HEADING "Gender";

COLUMN Branch\_ID FORMAT A9 HEADING "Branch ID";

COLUMN Room\_booking\_ID FORMAT A15 HEADING "Room Booking ID";

COLUMN Staff\_ID FORMAT A8 HEADING "Staff ID";

COLUMN Staff\_name FORMAT A15 HEADING "Staff Name";

TTITLE CENTER 'Staff handle rooms booking for &B\_ID students' \_DATE -

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON Staff\_name on Staff\_ID on branch\_ID SKIP 1 ON Student\_ID

SELECT SF.Staff\_ID,Staff\_name,S.Branch\_ID,S.Student\_ID,Student\_name,Student\_contact,Student\_gender,Room\_booking\_ID

FROM Student S,Branch B,Room\_booking R,Staff SF

WHERE S.Branch\_ID=B.Branch\_ID AND R.Student\_ID=S.Student\_ID AND R.Staff\_ID=SF.Staff\_ID AND B.Branch\_ID='&B\_ID'

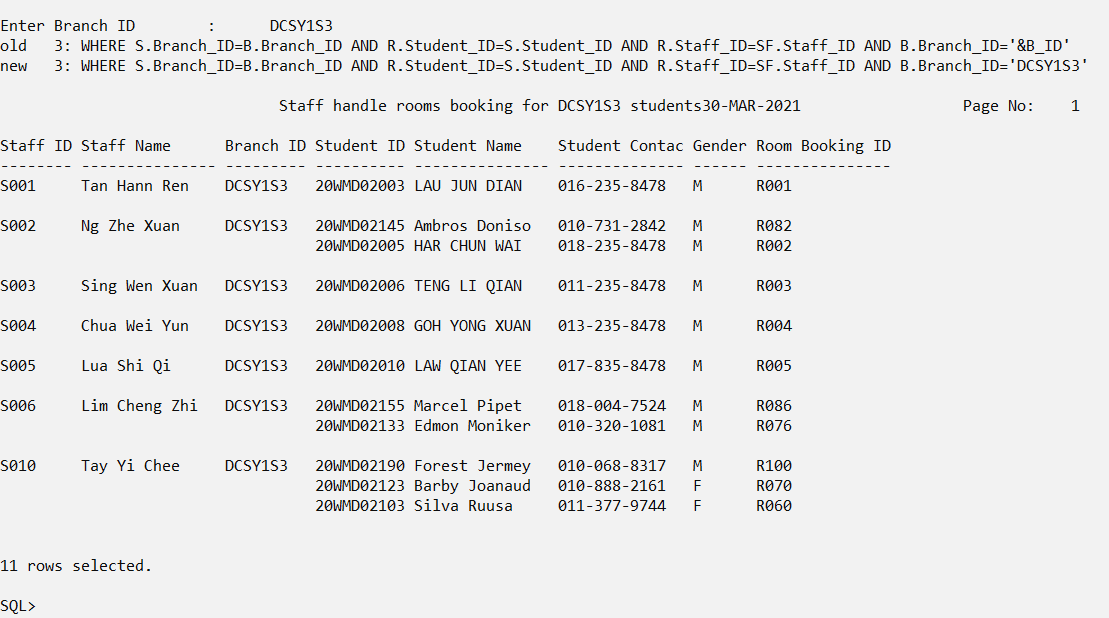
ORDER BY Staff\_ID;

CLEAR COLUMNS

CLEAR BREAKS

TTITLE OFF

Sample Output: Screenshot



### **6.5.3 Query 3: Detail report of the branch student borrowed which book**

PURPOSE: Let the user enter the branch ID to display the books borrowed by students of the branch to be viewed.

IMPORTANCE: To know what books the students of which branch have borrowed.

SQL Statement:

SET linesize 120

SET pagesize 100

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY';

cl scr

ACCEPT Branch\_ID char FORMAT 'A8' PROMPT 'Enter Branch ID : '

COLUMN Student\_ID FORMAT A10 HEADING "Student ID ";

COLUMN Student\_name FORMAT A15 HEADING "Student Name";

COLUMN Branch\_ID FORMAT A9 HEADING "Branch ID";

COLUMN book\_id FORMAT A7 HEADING "Book ID";

COLUMN book\_title FORMAT A25 HEADING "Book Title";

TTITLE CENTER 'Student &Branch\_ID borrowing books list' \_DATE -

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

BREAK ON book\_id on book\_title on branch\_id SKIP 1 ON Student\_ID

SELECT Branch\_ID,B.book\_id,book\_title,S.Student\_ID,Student\_name

FROM Student S,Book\_loan\_details BLD,Book B

WHERE S.Student\_ID=BLD.Student\_ID AND Branch\_ID='&Branch\_ID' AND B.book\_id=BLD.book\_id

ORDER BY book\_id;

CLEAR COLUMNS

CLEAR BREAK

TTITLE OFF

Sample Output: Screenshot

