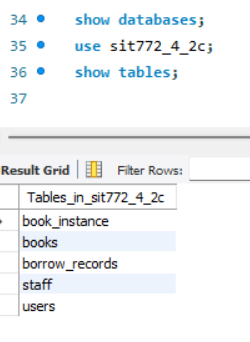
# Task

## Write SQL statements to create all tables in the final normalized ERD in Task 4.2C

|  |
| --- |
| CREATE TABLE USERS (  USER\_ID NUMERIC ( 20, 0 ) PRIMARY KEY,  FIRST\_NAME VARCHAR ( 25 ) NOT NULL,  SURNAME VARCHAR ( 25 ) NOT NULL,  `PASSWORD` VARCHAR ( 50 ) NOT NULL,  PHONE CHAR ( 10 ) NOT NULL,  EMAIL VARCHAR ( 100 ),  BIRTHDAY DATE,  GENDER CHAR ( 1 ) DEFAULT '1' COMMENT '1 MALE, 2 FEMALE',  `LANGUAGE` CHAR ( 3 ) DEFAULT '1' COMMENT '1 ENGLISH, 2 CHINESE, 3 OTHER',  ADDR VARCHAR ( 50 ) NOT NULL,  SUBURB VARCHAR ( 10 ) NOT NULL,  STATE VARCHAR ( 5 ) NOT NULL,  POSTCODE CHAR ( 4 ) NOT NULL,  COUNTRY VARCHAR ( 10 ),  RESIDENTIAL\_AREA VARCHAR ( 10 )  ); |
| CREATE TABLE STAFF (  STAFF\_ID NUMERIC ( 20, 0 ) PRIMARY KEY,  `NAME` VARCHAR ( 50 ) NOT NULL,  PHONE CHAR ( 10 ) NOT NULL,  BIRTHDAY DATE,  EMAIL VARCHAR ( 100 ),  `LANGUAGE` CHAR ( 3 ) DEFAULT '1' COMMENT '1 ENGLISH, 2 CHINESE, 3 OTHER',  ADDR VARCHAR ( 50 ) NOT NULL,  SUBURB VARCHAR ( 10 ) NOT NULL,  STATE VARCHAR ( 5 ) NOT NULL,  POSTCODE CHAR ( 4 ) NOT NULL,  COUNTRY VARCHAR ( 10 ),  RESIDENTIAL\_AREA VARCHAR ( 10 )  ); |
| CREATE TABLE BOOKS (  BOOK\_ID NUMERIC ( 20, 0 ) PRIMARY KEY,  BOOK\_NAME VARCHAR ( 100 ) NOT NULL,  MAIN\_TITLE VARCHAR ( 200 ) NOT NULL,  AUTHOR VARCHAR ( 50 ) NOT NULL,  WORK VARCHAR ( 100 ),  IMPRINT VARCHAR ( 100 ) NOT NULL,  COLLATION VARCHAR ( 100 ),  AUDIENCE VARCHAR ( 200 ),  ISBN VARCHAR ( 50 ),  DEWEY\_CLASS VARCHAR ( 150 ),  `LANGUAGE` CHAR ( 3 ) DEFAULT '1' COMMENT '1 ENGLISH, 2 CHINESE, 3 OTHER',  `SUBJECT` VARCHAR ( 50 ) NOT NULL,  BRN VARCHAR ( 100 ) NOT NULL,  BOOKMARK\_LINK VARCHAR ( 255 ),  CREATOR NUMERIC ( 20, 0 ) NOT NULL,  UPDATER NUMERIC ( 20, 0 ) NOT NULL,  FOREIGN KEY ( CREATOR ) REFERENCES STAFF ( STAFF\_ID ),  FOREIGN KEY ( UPDATER ) REFERENCES STAFF ( STAFF\_ID )  ); |
| CREATE TABLE BOOK\_INSTANCE (  INSTANCE\_ID NUMERIC ( 30, 0 ) PRIMARY KEY,  BOOK\_ID NUMERIC ( 20, 0 ),  IS\_PHYSICAL\_BOOK CHAR ( 1 ) DEFAULT 'Y' COMMENT 'Y, N',  LOCATION VARCHAR ( 100 ),  LINK VARCHAR ( 200 ) ,  FOREIGN KEY (BOOK\_ID) REFERENCES BOOKS(BOOK\_ID)  ); |
| CREATE TABLE BORROW\_RECORDS (  BORROW\_ID NUMERIC ( 30, 0 ) PRIMARY KEY,  INSTANCE\_ID NUMERIC ( 30, 0 ),  USER\_ID NUMERIC ( 30, 0 ),  START\_DATE DATE,  DUE\_DATE DATE,  IS\_FINISH CHAR ( 1 ) DEFAULT 'Y' COMMENT 'Y, N',  FOREIGN KEY (INSTANCE\_ID) REFERENCES BOOK\_INSTANCE(INSTANCE\_ID),  FOREIGN KEY (USER\_ID) REFERENCES USERS(USER\_ID)  ); |



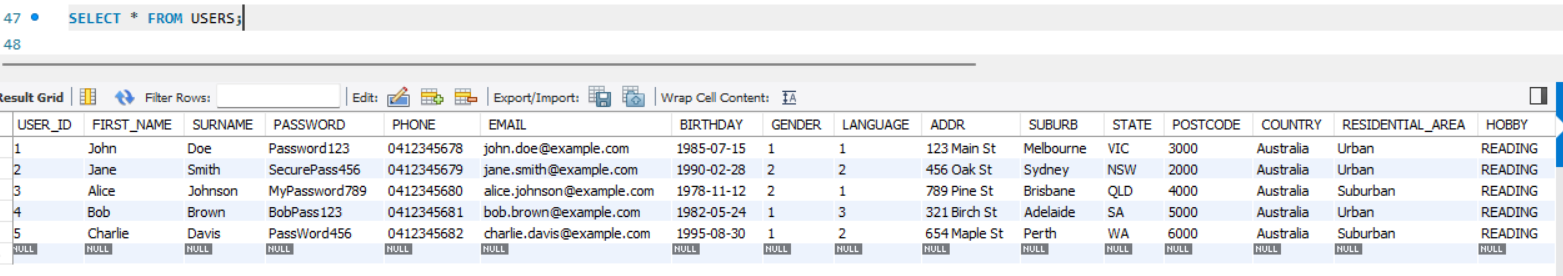
## Insert at least FIVE records in each table using SQL.

|  |
| --- |
| INSERT INTO USERS (  USER\_ID, FIRST\_NAME, SURNAME, `PASSWORD`, PHONE, EMAIL, BIRTHDAY, GENDER, `LANGUAGE`, ADDR, SUBURB, STATE, POSTCODE, COUNTRY, RESIDENTIAL\_AREA  ) VALUES  (1, 'John', 'Doe', 'Password123', '0412345678', 'john.doe@example.com', '1985-07-15', '1', '1', '123 Main St', 'Melbourne', 'VIC', '3000', 'Australia', 'Urban'),  (2, 'Jane', 'Smith', 'SecurePass456', '0412345679', 'jane.smith@example.com', '1990-02-28', '2', '2', '456 Oak St', 'Sydney', 'NSW', '2000', 'Australia', 'Urban'),  (3, 'Alice', 'Johnson', 'MyPassword789', '0412345680', 'alice.johnson@example.com', '1978-11-12', '2', '1', '789 Pine St', 'Brisbane', 'QLD', '4000', 'Australia', 'Suburban'),  (4, 'Bob', 'Brown', 'BobPass123', '0412345681', 'bob.brown@example.com', '1982-05-24', '1', '3', '321 Birch St', 'Adelaide', 'SA', '5000', 'Australia', 'Urban'),  (5, 'Charlie', 'Davis', 'PassWord456', '0412345682', 'charlie.davis@example.com', '1995-08-30', '1', '2', '654 Maple St', 'Perth', 'WA', '6000', 'Australia', 'Suburban'); |
| INSERT INTO STAFF (  STAFF\_ID, `NAME`, PHONE, BIRTHDAY, EMAIL, `LANGUAGE`, ADDR, SUBURB, STATE, POSTCODE, COUNTRY, RESIDENTIAL\_AREA  ) VALUES  (1, 'Michael Green', '0412345673', '1975-03-15', 'michael.green@example.com', '1', '10 Elm St', 'Melbourne', 'VIC', '3000', 'Australia', 'Urban'),  (2, 'Lisa White', '0412345674', '1983-06-22', 'lisa.white@example.com', '2', '22 Cedar St', 'Sydney', 'NSW', '2000', 'Australia', 'Urban'),  (3, 'James Black', '0412345675', '1992-09-08', 'james.black@example.com', '1', '15 Willow St', 'Brisbane', 'QLD', '4000', 'Australia', 'Suburban'),  (4, 'Emily Brown', '0412345676', '1988-12-19', 'emily.brown@example.com', '3', '7 Ash St', 'Adelaide', 'SA', '5000', 'Australia', 'Urban'),  (5, 'Robert Harris', '0412345677', '1980-01-30', 'robert.harris@example.com', '1', '5 Spruce St', 'Perth', 'WA', '6000', 'Australia', 'Suburban'); |
| INSERT INTO BOOKS (  BOOK\_ID, BOOK\_NAME, MAIN\_TITLE, AUTHOR, WORK, IMPRINT, COLLATION, AUDIENCE, ISBN, DEWEY\_CLASS, `LANGUAGE`, `SUBJECT`, BRN, BOOKMARK\_LINK, CREATOR, UPDATER  ) VALUES  (30000000000000000001, 'The Great Adventure', 'The Great Adventure: A Journey Through the Unknown', 'John Smith', 'Adventure Works', 'Penguin Books', '350 pages; 20 cm', 'Young Adults', '978-3-16-148410-0', '823/.92', '1', 'Adventure', 'BRN001', 'http://example.com/book1', 1, 2),  (30000000000000000002, 'Understanding Science', 'Understanding Science: Concepts and Applications', 'Jane Doe', 'Science Textbooks', 'Oxford University Press', '500 pages; 24 cm', 'Students and Professionals', '978-1-23-456789-0', '500', '1', 'Science', 'BRN002', 'http://example.com/book2', 3, 1),  (30000000000000000003, 'World History', 'World History: From Ancient to Modern Times', 'Alice Johnson', 'History Publications', 'HarperCollins', '450 pages; 23 cm', 'General Audience', '978-0-12-345678-9', '909', '1', 'History', 'BRN003', 'http://example.com/book3', 2, 4),  (30000000000000000004, 'The Art of Coding', 'The Art of Coding: Mastering Programming Languages', 'Robert Brown', 'Tech Works', 'McGraw-Hill', '600 pages; 25 cm', 'Students and Developers', '978-4-56-789012-3', '005.1', '1', 'Technology', 'BRN004', 'http://example.com/book4', 4, 3),  (30000000000000000005, 'Culinary Delights', 'Culinary Delights: Recipes from Around the World', 'Emily White', 'Gourmet Publishing', 'Random House', '300 pages; 22 cm', 'General Audience', '978-7-89-012345-6', '641.5', '1', 'Cooking', 'BRN005', 'http://example.com/book5', 5, 2); |
| INSERT INTO BOOK\_INSTANCE (  INSTANCE\_ID, BOOK\_ID, IS\_PHYSICAL\_BOOK, LOCATION, LINK  ) VALUES  (400000000000000000000000001, 30000000000000000001, 'Y', 'Main Library - Shelf A3', NULL),  (400000000000000000000000002, 30000000000000000002, 'N', NULL, 'http://example.com/ebook2'),  (400000000000000000000000003, 30000000000000000003, 'Y', 'Main Library - Shelf B2', NULL),  (400000000000000000000000004, 30000000000000000004, 'N', NULL, 'http://example.com/ebook4'),  (400000000000000000000000005, 30000000000000000005, 'Y', 'Main Library - Shelf C1', NULL); |
| INSERT INTO BORROW\_RECORDS (  BORROW\_ID, INSTANCE\_ID, USER\_ID, START\_DATE, DUE\_DATE, IS\_FINISH  ) VALUES  (50000000000000000000000000001, 400000000000000000000000001, 1, '2024-08-01', '2024-08-15', 'Y'),  (50000000000000000000000000002, 400000000000000000000000002, 2, '2024-07-20', '2024-08-03', 'N'),  (50000000000000000000000000003, 400000000000000000000000003, 3, '2024-08-10', '2024-08-24', 'Y'),  (50000000000000000000000000004, 400000000000000000000000004, 4, '2024-08-05', '2024-08-19', 'N'),  (50000000000000000000000000005, 400000000000000000000000005, 5, '2024-08-12', '2024-08-26', 'Y'); |

## Write and run an SQL statement to add one additional column (attribute) in any one of the existing tables with a default value.

Add a new column in USERS table, column name is hobby, data type is varchar, length is 20, default value is READING;

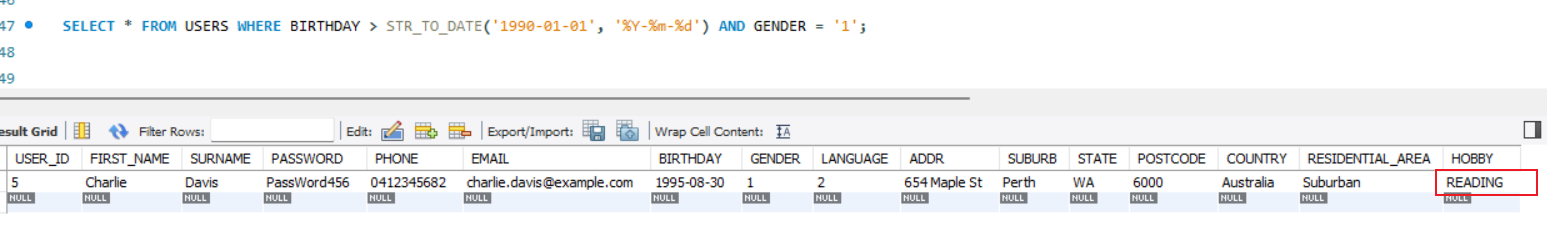
SQL: ALTER TABLE USERS ADD COLUMN HOBBY VARCHAR(25) NULL DEFAULT 'READING' AFTER `RESIDENTIAL\_AREA`;



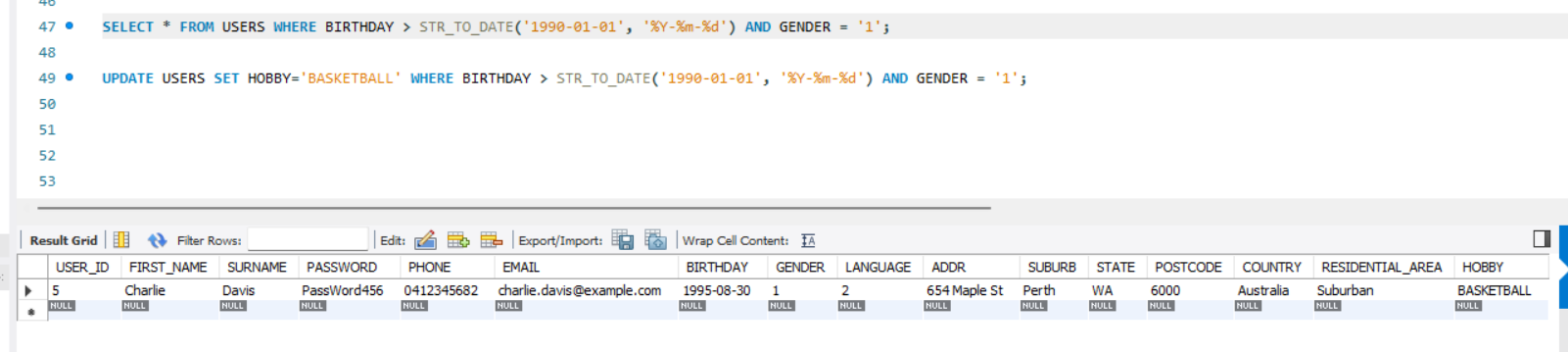
## Write and run an SQL statement to update the default value of the newly added column to a different value for certain rows based on a condition using any other column.

Change the hobby of users who were born after 1990 and whose gender is male to BASKETBALL.

SELECT \* FROM USERS WHERE BIRTHDAY > STR\_TO\_DATE('1990-01-01', '%Y-%m-%d') AND GENDER = '1';



UPDATE USERS SET HOBBY='BASKETBALL' WHERE BIRTHDAY > STR\_TO\_DATE('1990-01-01', '%Y-%m-%d') AND GENDER = '1';



## In this section, you are required to write SQL queries to interact with the database you implemented. Answer each SQL question in this section with the following:

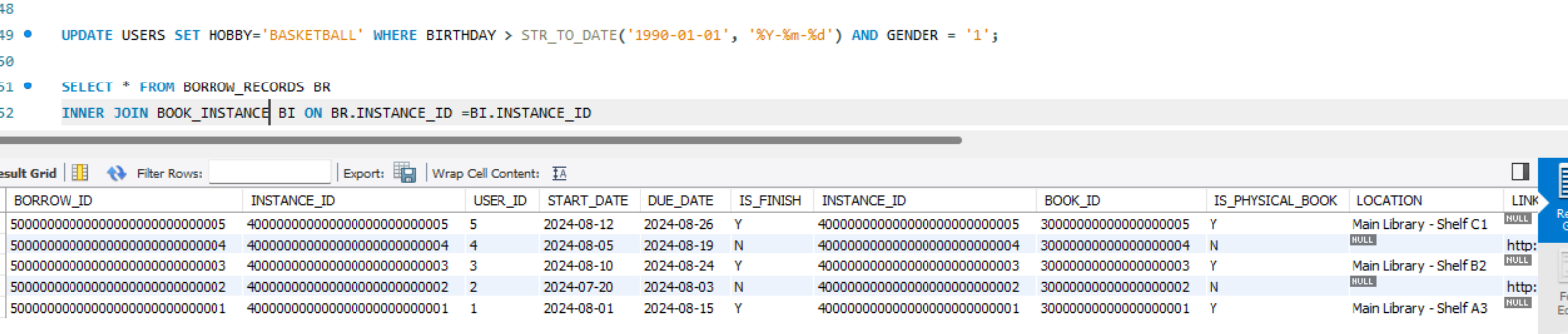
### a) Write an SQL query to demonstrate the use of SELECT with INNER JOIN and ORDER BY.

**SELECT** \* **FROM** BORROW\_RECORDS BR

**INNER** **JOIN** BOOK\_INSTANCE BI **ON** BR.INSTANCE\_ID =BI.INSTANCE\_ID

**INNER** **JOIN** BOOKS B **ON** BI.BOOK\_ID =B.BOOK\_ID

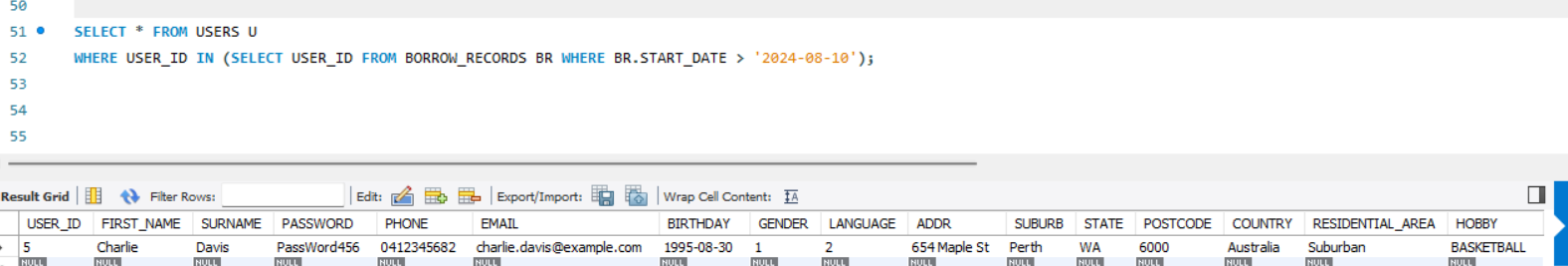
**ORDER** **BY** BORROW\_ID **DESC** ;



### b) Write an SQL query to demonstrate the use of SELECT with WHERE and IN.

SELECT \* FROM USERS U

WHERE USER\_ID IN (SELECT USER\_ID FROM BORROW\_RECORDS BR WHERE BR.START\_DATE > '2024-08-10');



### c) Write an SQL query to demonstrate the use of at least one DATE function.

SELECT CONCAT(U.FIRST\_NAME,' ',U.SURNAME) 'USER\_NAME', B.BOOK\_NAME ,DATEDIFF(NOW() ,DUE\_DATE) AS 'OVERDUE'

FROM BORROW\_RECORDS BR,BOOK\_INSTANCE BI ,BOOKS B ,USERS U

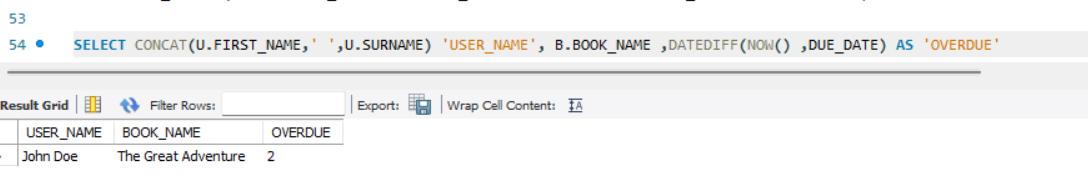
WHERE

BR.INSTANCE\_ID =BI.INSTANCE\_ID

AND BI.BOOK\_ID =B.BOOK\_ID

AND BR.USER\_ID =U.USER\_ID

AND U.FIRST\_NAME ='JOHN';



### d) Write an SQL statement to create a VIEW using a SELECT statement with a JOIN. Provide the statement to create the VIEW you want and demonstrate the output of the VIEW using ‘SELECT \* FROM ;’.

*CREATE VIEW USER\_BOOK\_OVERDUE AS  
SELECT* U.*USER\_ID*, CONCAT(U.*FIRST\_NAME*,' ',U.*SURNAME*), B.*BOOK\_NAME* ,DATEDIFF(NOW() ,*DUE\_DATE*) *AS* 'OVERDUE'   
*FROM BORROW\_RECORDS BR*,*BOOK\_INSTANCE BI* ,*BOOKS* B ,*USERS* U  
*WHERE  
BR*.*INSTANCE\_ID* =*BI*.*INSTANCE\_ID   
AND BI*.*BOOK\_ID* =B.*BOOK\_ID   
AND BR*.*USER\_ID* =U.*USER\_ID* ;  
  
*SELECT* \* *FROM USER\_BOOK\_OVERDUE*;

