AIT-524-003

Homework Assignment 12

ER Diagram:

Student Housing and Residence Life Database

**A screenshot of a computer

Description automatically generated**

1. Write an SQL query that uses a single-row subquery in a WHERE clause. Explain what the query is intended to do.

***Retrieve the dorm name, dorm location, dorm price where the dorm price is greater than average dorm price.***

A screen shot of a social media post

Description automatically generated

2. Write an SQL query that uses a multiple-column subquery in a FROM clause. Explain what the query is intended to do.

***Retrieve dorm name, dorm type, dorm price where the dorm price is equal to the average dorm price.***

A close up of text on a black background

Description automatically generated

3. Write an SQL query that is based on multiple tables tables and uses a subquery with the GROUP BY statement and HAVING clause. Explain what the query is intended to do.

***Retrieve dorm room number and supervisor id where average dorm room number is less than average supervisor id for dorm room number 7777 categorized by dorm room number and supervisor id.***

A screenshot of a computer

Description automatically generated

4. Write an SQL query that is based on multiple tables and uses a multiple-row subquery in a WHERE clause. The subquery will include the GROUP BY statement and another multiple-row subquery in a HAVING clause. Explain what the query is intended to do.

***Retrieve student gender, firstname and the service id he is given based on the student category.***

A screenshot of a computer screen

Description automatically generated

5. Write an SQL query that joins three tables and uses any type of a subquery. Explain what the query is intended to do.

***Retrieve student first name, last name and service name that has been done that includes characters ‘Rep’.***

A picture containing indoor

Description automatically generated

6. Write an SQL query that is based on multiple tables and uses the DECODE function. Explain what the query is intended to do.

***Retrieve supervisor first name, last name, dorm room number, dorm type where he has worked based on dorm type and dorm price.***

A screenshot of a computer screen

Description automatically generated

7. Create the Faculty table and populate it with data using the script below:   
  
CREATE TABLE faculty (f\_id NUMBER(6), f\_last VARCHAR2(30) ,f\_first VARCHAR2(30), f\_mi CHAR(1), loc\_id NUMBER(5), f\_phone VARCHAR2(10), f\_rank VARCHAR2(9), f\_super NUMBER(6), CONSTRAINT faculty\_f\_id\_pk PRIMARY KEY(f\_id));  
INSERT INTO faculty VALUES (1, 'Marx', 'Teresa', 'J', 9, '4075921695', 'Associate', 4);  
INSERT INTO faculty VALUES (2, 'Zhulin', 'Mark', 'M', 10, '4073875682', 'Full', NULL);  
INSERT INTO faculty VALUES (3, 'Langley', 'Colin', 'A', 12, '4075928719', 'Assistant', 4);  
INSERT INTO faculty VALUES (4, 'Brown', 'Jonnel', 'D', 11, '4078101155', 'Full', NULL);

Check the result using the select \* from faculty; command.

A screenshot of a cell phone

Description automatically generated

8. Create the Bonus table that consists of two columns: f\_id (PK) and bonus. For the f\_id column, use the same description as in the Faculty table. For the bonus column, use the NUMBER data type and the DEFAULT constraint to set the values for the bonus column to 1000 (bonus amount). Next, use a subquery to copy ids of mentors given in the Faculty table into the Bonus table.

Check the result using the select \* from bonus; command.

A screen shot of a computer

Description automatically generated

9. Add two new records to the Faculty table using the command below. These records represent new faculty who came to the university this year.  
  
INSERT INTO faculty VALUES (5, 'Sealy', 'James', 'L', 13, '4079817153', 'Associate', 1);  
INSERT INTO faculty VALUES (6, 'Smith', 'John', 'D', 10, '4238102345', 'Full', NULL);

Check the result using the select \* from faculty; command.

A screenshot of a computer screen

Description automatically generated

10. Assume that the same Bonus table is used next year to assign and update bonuses. Use the MERGE statement to modify the Bonus table as follows:

- if a mentor already exists in the Bonus table, increase the bonus by 1%

- If there is a new mentor in the Faculty table, add him/her to the BONUS table

Check the result using the select \* from bonus; command.

A screen shot of a smart phone

Description automatically generated