Instructions

For <u>each</u> problem:

- **Write and execute an SQL query in Oracle Live SQL or SQL*Plus**
- **Execute the following command: select sysdate, 'your name' from dual; where your_name is substituted with your name**
- **Take a screenshot that includes both SQL statements and all results**
- **Copy and paste the screenshot into a Word file containing your solutions**

Assignment:

For the database you designed in the previous assignments, complete the following problems:

- **Write an SQL query that uses a single-row subquery in a WHERE clause. Explain what the query is intended to do.**
- **Write an SQL query that uses a multiple-column subquery in a FROM clause. Explain what the query is intended to do.**
- **Write an SQL query that is based on multiple tables and uses a subquery with the GROUP BY statement and HAVING clause. Explain what the query is intended to do.**
- **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.**
- **Write an SQL query that joins three tables and uses any type of a subquery. Explain what the query is intended to do.**
- **Write an SQL query that is based on multiple tables and uses the DECODE function. Explain what the query is intended to do.**

For Problems 7-10 below, consider the following business scenario:

> **A university wants to keep track of bonuses given to the professors who mentor (supervise) junior faculty members. Data about all professors is available in the Faculty table (see below). In that table, the f_super column represents a faculty id of a mentor (if any). A separate Bonus table is needed to assign and keep track of bonuses. The Bonus table will have a default bonus of 1000. It will be updated once a year to add new mentors and to update bonuses for the existing ones.**

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.

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.

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.

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.

Note: Include your ERD/EERD from your previous assignment.

My Session

Actions | Reset Session | Save

Statement 229

```
SELECT title, retail
FROM books
WHERE retail < (SELECT MAX(retail) FROM books)
```

TITLE	RETAIL
BODYBUILD IN 10 MINUTES A DAY	30.95
REVENGE OF MICKEY	22
BUILDING A CAR WITH TOOTHPICKS	59.95
DATABASE IMPLEMENTATION	55.95
COOKING WITH MUSHROOMS	19.95
HOLY GRAIL OF ORACLE	75.95
HANDCRANKED COMPUTERS	25
E-BUSINESS THE EASY WAY	54.5
THE WOK WAY TO COOK	28.75
BIG BEAR AND LITTLE DOVE	8.95
HOW TO GET FASTER PIZZA	29.95
HOW TO MANAGE THE MANAGER	31.95
SHORTEST POEMS	39.95

Download CSV

Statement 230

```
SELECT title, retail
FROM books
WHERE retail < (SELECT MAX(retail) FROM books)
```

TITLE	RETAIL
BODYBUILD IN 10 MINUTES A DAY	30.95
REVENGE OF MICKEY	22
BUILDING A CAR WITH TOOTHPICKS	59.95
DATABASE IMPLEMENTATION	55.95
COOKING WITH MUSHROOMS	19.95
HOLY GRAIL OF ORACLE	75.95
HANDCRANKED COMPUTERS	25
E-BUSINESS THE EASY WAY	54.5
THE WOK WAY TO COOK	28.75
BIG BEAR AND LITTLE DOVE	8.95
HOW TO GET FASTER PIZZA	29.95
HOW TO MANAGE THE MANAGER	31.95
SHORTEST POEMS	39.95

Download CSV

Statement **231**

```
SELECT b1.category, avg_retail
FROM (SELECT category, AVG(retail) AS avg_retail
      FROM books
      GROUP BY category) b1
WHERE avg_retail > 20
```

CATEGORY	AVG_RETAIL
CHILDREN	34.45
LITERATURE	39.95
FAMILY LIFE	55.975
COMPUTER	52.85
COOKING	24.35
FITNESS	30.95
BUSINESS	31.95
SELF HELP	29.95

Download CSV

8 rows selected.

Statement **232**

```
SELECT title, retail
FROM books
WHERE retail < (SELECT MAX(retail) FROM books)
```

Statement **232**

```
SELECT title, retail
FROM books
WHERE retail < (SELECT MAX(retail) FROM books)
```

TITLE	RETAIL
BODYBUILD IN 10 MINUTES A DAY	30.95
REVENGE OF MICKEY	22
BUILDING A CAR WITH TOOTHPICKS	59.95
DATABASE IMPLEMENTATION	55.95
COOKING WITH MUSHROOMS	19.95
HOLY GRAIL OF ORACLE	75.95
HANDCRANKED COMPUTERS	25
E-BUSINESS THE EASY WAY	54.5
THE WOK WAY TO COOK	28.75
BIG BEAR AND LITTLE DOVE	8.95
HOW TO GET FASTER PIZZA	29.95
HOW TO MANAGE THE MANAGER	31.95
SHORTEST POEMS	39.95

Download CSV

Statement **233**

```
SELECT b1.category, avg_retail
FROM (SELECT category, AVG(retail) AS avg_retail
      FROM books
      GROUP BY category) b1
WHERE avg_retail > 20
```

CATEGORY	AVG_RETAIL
CHILDREN	34.45
LITERATURE	39.95
FAMILY LIFE	55.975
COMPUTER	52.85
COOKING	24.35
FITNESS	30.95
BUSINESS	31.95
SELF HELP	29.95

Download CSV

8 rows selected.

My Session

Actions | Reset Session

8 rows selected.

Statement **234**

```
SELECT category, AVG(retail) AS avg_price
FROM books
GROUP BY category
HAVING AVG(retail) > (SELECT AVG(retail) FROM books)
```

CATEGORY	AVG_PRICE
FAMILY LIFE	55.975
COMPUTER	52.85

Download CSV

2 rows selected.

Statement **235**

```
SELECT sysdate, 'Chrissie Raj' FROM dual
```

SYSDATE	'CHRISSIERAJ'
17-DEC-24	Chrissie Raj

Download CSV

Statement **236**

```
SELECT title, pubid
FROM books
WHERE pubid IN (
    SELECT pubid
    FROM books
```

7219205801::::

My Session

Actions | Reset Session | Save

Statement 236

```
SELECT title, pubid
FROM books
WHERE pubid IN (
    SELECT pubid
    FROM books
    GROUP BY pubid
    HAVING COUNT(*) > 2 AND AVG(retail) > 25
)
```

TITLE	PUBID
BODYBUILD IN 10 MINUTES A DAY	4
DATABASE IMPLEMENTATION	3
COOKING WITH MUSHROOMS	4
HOLY GRAIL OF ORACLE	3
HANDCRANKED COMPUTERS	3
PAINLESS CHILD-REARING	5
THE WOK WAY TO COOK	4
BIG BEAR AND LITTLE DOVE	5
HOW TO GET FASTER PIZZA	4
SHORTEST POEMS	5

Download CSV

10 rows selected.

My Session

Actions | Reset Session | Save

Statement 237

```
SELECT sysdate, 'Chrissie Raj' FROM dual
```

SYSDATE	'CHRISSIERAJ'
17-DEC-24	Chrissie Raj

Download CSV

Statement 238

```
SELECT pubid, COUNT(*) AS book_count
FROM books
GROUP BY pubid
HAVING SUM(retail) > (SELECT AVG(retail) * 2 FROM books)
```

PUBID	BOOK_COUNT
2	2
4	4
5	3
3	3

Download CSV

4 rows selected.

Statement 239

```
SELECT sysdate, 'Chrissie Raj' FROM dual
```

My Session

Actions | Reset Session | Save

Statement 239

```
SELECT sysdate, 'Chrissie Raj' FROM dual
```

SYSDATE	'CHRISSIERAJ'
17-DEC-24	Chrissie Raj

Download CSV

Statement 240

```
SELECT o.order#, o.shipstate, oi.quantity, b.title
FROM orders o
JOIN orderitems oi ON o.order# = oi.order#
JOIN books b ON oi.isbn = b.isbn
WHERE o.order# IN (
    SELECT order#
    FROM orderitems
    GROUP BY order#
    HAVING SUM(quantity * paideach) > 100
)
```

ORDER#	SHIPSTATE	QUANTITY	TITLE
1001	GA	1	HOW TO MANAGE THE MANAGER
1001	GA	1	PAINLESS CHILD-REARING
1002	IL	2	DATABASE IMPLEMENTATION
1003	FL	1	DATABASE IMPLEMENTATION
1003	FL	1	BODYBUILD IN 10 MINUTES A DAY

My Session

Actions | Reset Session | Save

1004	NJ	2	PAINLESS CHILD-REARING
1007	TX	3	HOLY GRAIL OF ORACLE
1007	TX	1	E-BUSINESS THE EASY WAY
1007	TX	1	BIG BEAR AND LITTLE DOVE
1007	TX	1	DATABASE IMPLEMENTATION
1012	MI	1	BIG BEAR AND LITTLE DOVE
1012	MI	2	HANDCRANKED COMPUTERS
1012	MI	1	PAINLESS CHILD-REARING
1012	MI	1	REVENGE OF MICKEY

Download CSV

15 rows selected.

Statement 241

```
SELECT sysdate, 'Chrissie Raj' FROM dual
```

SYSDATE	'CHRISSIERAJ'
17-DEC-24	Chrissie Raj

Download CSV

Statement 242

```
SELECT title,
       DECODE(
```

My Session

Actions | Reset Session | Save

Statement 242

```
SELECT title,
       DECODE(
           SIGN(retail - 30),
           1, 'Expensive',
           0, 'Moderate',
           -1, 'Cheap'
       ) AS price_category
FROM books
```

TITLE	PRICE_CATEGORY
BODYBUILD IN 10 MINUTES A DAY	Expensive
REVENGE OF MICKEY	Cheap
BUILDING A CAR WITH TOOTHPICKS	Expensive
DATABASE IMPLEMENTATION	Expensive
COOKING WITH MUSHROOMS	Cheap
HOLY GRAIL OF ORACLE	Expensive
HANDCRANKED COMPUTERS	Cheap
E-BUSINESS THE EASY WAY	Expensive
PAINLESS CHILD-REARING	Expensive
THE WOK WAY TO COOK	Cheap
BIG BEAR AND LITTLE DOVE	Cheap
HOW TO GET FASTER PIZZA	Cheap

My Session

Actions | Reset Session | Save

Statement 243

```
SELECT sysdate, 'Chrissie Raj' FROM dual
```

SYSDATE	'CHRISSIERAJ'
17-DEC-24	Chrissie Raj

Download CSV

Statement 244

```
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)
)
```

Table created.

Statement 245

```
INSERT INTO faculty VALUES (1, 'Marx', 'Teresa', 'J', 9, '4075921695', 'Associate', 4)
```

1 row(s) inserted.

Statement 246

```
INSERT INTO faculty VALUES (2, 'Zhulin', 'Mark', 'M', 10, '4073875682', 'Full', NULL)
```

My Session

Actions | Reset Session | Sa

1 row(s) inserted.

Statement 246

```
INSERT INTO faculty VALUES (2, 'Zhulin', 'Mark', 'M', 10, '4073875682', 'Full', NULL)
```

1 row(s) inserted.

Statement 247

```
INSERT INTO faculty VALUES (3, 'Langley', 'Colin', 'A', 12, '4075928719', 'Assistant', 4)
```

1 row(s) inserted.

Statement 248

```
INSERT INTO faculty VALUES (4, 'Brown', 'Jonnel', 'D', 11, '4078101155', 'Full', NULL)
```

1 row(s) inserted.

Statement 249

```
SELECT * FROM faculty
```

F_ID	F_LAST	F_FIRST	F_MI	LOC_ID	F_PHONE	F_RANK	F_SUPER
1	Marx	Teresa	J	9	4075921695	Associate	4
2	Zhulin	Mark	M	10	4073875682	Full	-
3	Langley	Colin	A	12	4075928719	Assistant	4
4	Brown	Jonnel	D	11	4078101155	Full	-

Download CSV

My Session

Actions | Reset Session | Save

4 rows selected.

Statement 250

```
SELECT sysdate, 'Chrissie Raj' FROM dual
```

SYSDATE	'CHRISSIERAJ'
17-DEC-24	Chrissie Raj

Download CSV

Statement 251

```
CREATE TABLE bonus (
    f_id NUMBER(6) PRIMARY KEY,
    bonus NUMBER DEFAULT 1000
)
```

Table created.

Statement 252

```
INSERT INTO bonus (f_id)
SELECT DISTINCT f_super
FROM faculty
WHERE f_super IS NOT NULL
```

1 row(s) inserted.

Statement 253

```
SELECT * FROM bonus
```


My Session
Actions
Reset Session
Save
Statement 253
SELECT * FROM bonus
F_ID
BONUS
4
1000
Download CSV
Statement 254
SELECT sysdate, 'Chrissie Raj' FROM dual
SYSDATE
'CHRISSIERAJ'
17-DEC-24
Chrissie Raj
Download CSV
Statement 255
INSERT INTO faculty VALUES (5, 'Sealy', 'James', 'L', 13, '4079817153', 'Associate', 1)
1 row(s) inserted.
Statement 256
INSERT INTO faculty VALUES (6, 'Smith', 'John', 'D', 10, '4238102345', 'Full', NULL)
1 row(s) inserted.

My Session

Actions | Reset Session | Save

1 row(s) inserted.

Statement 257

```
SELECT * FROM faculty
```

F_ID	F_LAST	F_FIRST	F_MI	LOC_ID	F_PHONE	F_RANK	F_SUPER
1	Marx	Teresa	J	9	4075921695	Associate	4
2	Zhulin	Mark	M	10	4073875682	Full	-
3	Langley	Colin	A	12	4075928719	Assistant	4
4	Brown	Jonnel	D	11	4078101155	Full	-
5	Sealy	James	L	13	4079817153	Associate	1
6	Smith	John	D	10	4238102345	Full	-

Download CSV

6 rows selected.

Statement 258

```
SELECT sysdate, 'Chrissie Raj' FROM dual
```

SYSDATE	'CHRISSIERAJ'
17-DEC-24	Chrissie Raj

Download CSV

My Session

Actions | Reset Session | Save

Statement 259

```
MERGE INTO bonus b
USING (SELECT f_id FROM faculty WHERE f_super IS NOT NULL) f
ON (b.f_id = f.f_id)
WHEN MATCHED THEN
    UPDATE SET b.bonus = b.bonus * 1.01
WHEN NOT MATCHED THEN
    INSERT (f_id, bonus) VALUES (f.f_id, 1000)
```

Statement processed.

Statement 260

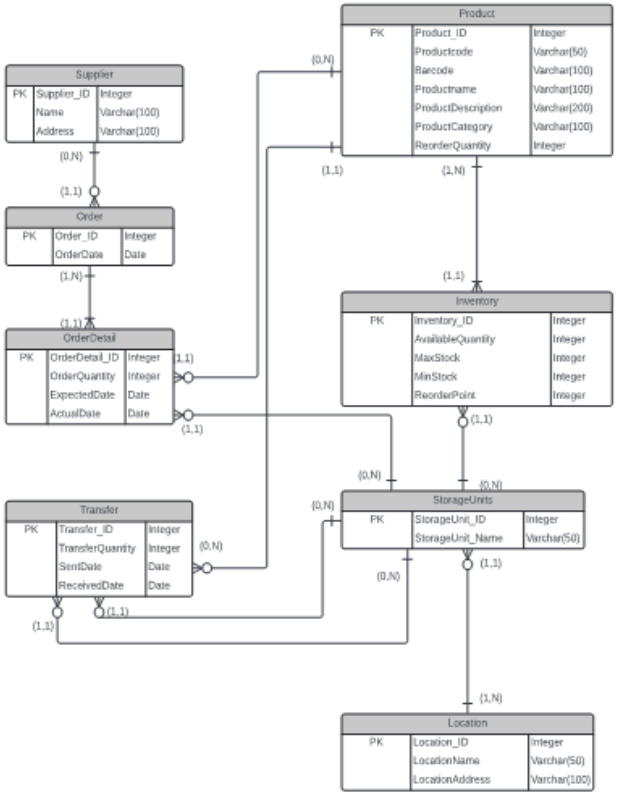
```
SELECT * FROM bonus
```

F_ID	BONUS
1	1000
3	1000
5	1000
4	1000

Download CSV

4 rows selected.

ERD:


Product
PK
Product_ID
Integer
ProductIcode
Varchar(50)
Barcode
Varchar(100)
Productname
Varchar(100)
ProductDescription
Varchar(200)
ProductCategory
Varchar(100)
ReorderQuantity
Integer
Supplier
PK
Supplier_ID
Integer
Name
Varchar(100)
Address
Varchar(100)
Order
PK
Order_ID
Integer
OrderDate
Date
Inventory
PK
Inventory_ID
Integer
AvailableQuantity
Integer
MaxStock
Integer
MinStock
Integer
ReorderPoint
Integer
OrderDetail
PK
OrderDetail_ID
Integer
OrderQuantity
Integer
ExpectedDate
Date
ActualDate
Date
Transfer
PK
Transfer_ID
Integer
TransferQuantity
Integer
SentDate
Date
ReceivedDate
Date
StorageUnits
PK
StorageUnit_ID
Integer
StorageUnit_Name
Varchar(50)
Location
PK
Location_ID
Integer
LocationName
Varchar(50)
LocationAddress
Varchar(100)
(0,N)
(1,1)
(1,N)