Data Base HW.2

کیان پورآذر ۴۰۱۳۱۴۰۳

سوال اول: الف:

```sql
CREATE TABLE Teams (
    team_id INT PRIMARY KEY,
    name VARCHAR(255),
    manager_id INT
);

CREATE TABLE Employees (
    emp_id INT PRIMARY KEY,
    name VARCHAR(255),
    position_id INT
);

CREATE TABLE TeamMembers (
    team_id INT,
    emp_id INT,
    PRIMARY KEY (team_id, emp_id)
);

CREATE TABLE Positions (
    position_id INT PRIMARY KEY,
    title VARCHAR(255),
    salary DECIMAL(10, 2)
);
```

ب:

```sql
SELECT DISTINCT e.name
FROM Employees e
JOIN Teams t ON e.emp_id = t.manager_id
JOIN Positions p ON e.position_id = p.position_id
WHERE e.emp_id IN (SELECT emp_id FROM TeamMembers)
  AND p.salary > 00100;
```

ج:

```sql
SELECT t.name AS team_name, e.name AS employee_name, p.salary
FROM Teams t
JOIN TeamMembers tm ON t.team_id = tm.team_id
JOIN Employees e ON tm.emp_id = e.emp_id
JOIN Positions p ON e.position_id = p.position_id
WHERE (t.team_id, p.salary) IN (
    SELECT tm.team_id, MAX(p.salary)
    FROM TeamMembers tm
    JOIN Employees e ON tm.emp_id = e.emp_id
    JOIN Positions p ON e.position_id = p.position_id
    GROUP BY tm.team_id
);
```

سوال دوم: الف:

```
CREATE VIEW HighValueSales AS
SELECT sale_id, product_name, sale_date, quantity_sold, price_per_unit,
       (quantity_sold * price_per_unit) AS total_sales
FROM Sales
WHERE (quantity_sold * price_per_unit) > 50000;
```

ب:

```
SELECT product_name, SUM(total_sales) AS total_sales_amount, SUM(quantity_sold) AS
total_quantity
FROM HighValueSales
GROUP BY product_name
HAVING SUM(total_sales) > 10000 AND SUM(quantity_sold) > 200
ORDER BY total_sales_amount DESC;
```

سوال سوم: الف:

نام کارمندانی را که در شرکت هایی نیویورک کار میکنند و حقوقشان از میانگین حقوق کارمندان شرکت "Future Innovations" بیشتر است.

ب:

اسم شرکت‌هایی را که کارمندانش حقوق بالای ۶۰۰۰۰ و بودجه پروژه‌شون بالای ۲۰۰۰۰۰ است.

سوال چهارم: الف:

```
SELECT dept_name, SUM(salary) AS total_salary
FROM instructor
GROUP BY dept_name
ORDER BY dept_name ASC;
```

ب:

```
FROM instructor
WHERE dept_name = 'Physics' AND salary BETWEEN 45000 AND 100000
ORDER BY salary ASC;
```

ج:

```
SELECT dept_name
FROM instructor
GROUP BY dept_name
HAVING SUM(salary) > 120000;
```

سوال پنجم:الف:

```
SELECT m.member_id, m.name, COUNT(b.borrow_id) AS books_borrowed
FROM member m
JOIN borrow b ON m.member_id = b.member_id
GROUP BY m.member_id, m.name
HAVING COUNT(b.borrow_id) > (
    SELECT AVG(borrow_count)
    FROM (
        SELECT COUNT(borrow_id) AS borrow_count
        FROM borrow
        GROUP BY member_id
    ) AS avg_borrow
);
```

ب:

```sql
SELECT bk.book_id, bk.title
FROM book bk
JOIN borrow br ON bk.book_id = br.book_id
GROUP BY bk.book_id, bk.title
HAVING COUNT(DISTINCT br.member_id) >= 2;
```

ج:

```sql
SELECT DISTINCT m.member_id, m.name
FROM member m
JOIN borrow br ON m.member_id = br.member_id
JOIN book bk ON br.book_id = bk.book_id
WHERE bk.price > (SELECT AVG(price) FROM book);
```

سوال ششم:

A.

```sql
SELECT select_list
FROM TableA AS a
LEFT JOIN TableB AS b
ON a.KEY = b.KEY
```

B.

```sql
SELECT select_list
FROM TableA AS a
LEFT JOIN TableB AS b
ON a.KEY = b.KEY
WHERE b.KEY IS NULL
```

C.

```sql
SELECT select_list
FROM TableA AS a
FULL OUTER JOIN TableB AS b
ON a.KEY = b.KEY
```

D.

```sql
SELECT select_list
FROM TableA AS a
FULL OUTER JOIN TableB AS b
ON a.KEY = b.KEY
WHERE a.KEY IS NULL
OR b.KEY IS NULL
```

E.

```sql
SELECT select_list
FROM TableA AS a
INNER JOIN TableB AS b
ON a.KEY = b.KEY
```

F.

```sql
SELECT select_list
FROM TableA AS a
RIGHT JOIN TableB AS b
ON a.KEY = b.KEY
```

G.

```
SELECT select_list
FROM TableA AS a
RIGHT JOIN TableB AS b
ON a.KEY = b.KEY
WHERE a.KEY IS NULL
```

سوال هفتم:

```
CREATE VIEW SalesmanLocations AS
SELECT salesman_id, name, city, commission
FROM Salesman;
```

سوال هشتم:

- Consists of a sequence of query and/or update statements.
- Atomic transaction
- Either fully executed or rolled back as if it never occurred
- Transactions begin implicitly and ended by one of the following:

- **Commit work** commits the current transaction
  •Making the updates performed by the transaction become permanent in the database.

  •After the transaction is committed, a new transaction is automatically started.
- **Rollback work** causes the current transaction to be rolled back
  •It undoes all the updates performed by the SQL statements in the transaction.

  •Thus, the database state is restored to what it was before the first statement of the transaction was executed.

سوال نهم:

```
GRANT SELECT(name, salary) ON instructor TO U1;

GRANT UPDATE(course_id, title, dept_name) ON course TO U2;

GRANT SELECT(course_id, title) ON course TO U3;
```