Data Base HW.2

کیان پورآذر ۴۰۱۳۱۴۰۳

سوال اول: الف:

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:

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:

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 ار 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:

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:

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.

SELECT select\_list

FROM TableA AS a

LEFT JOIN TableB AS b

ON a.KEY = b.KEY

B.

SELECT select\_list

FROM TableA AS a

LEFT JOIN TableB AS b

ON a.KEY = b.KEY

WHERE b.KEY IS NULL

C.

SELECT select\_list

FROM TableA AS a

FULL OUTER JOIN TableB AS b

ON a.KEY = b.KEY

D.

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.

SELECT select\_list

FROM TableA AS a

INNER JOIN TableB AS b

ON a.KEY = b.KEY

F.

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;