

گزارش آزمایش دوم – آزمایشگاه پایگاه داده

| | | |
|---------------|--------------------|-------------|
| Sailor | <u>Sailor_name</u> | Sailor_rank |
|---------------|--------------------|-------------|

| | | | |
|-------------|------------------|------------|-----------|
| Boat | <u>Boat_name</u> | Boat_color | Boat_rank |
|-------------|------------------|------------|-----------|

| | | | |
|----------------|--------------------|------------------|---------|
| Reserve | <u>Sailor_name</u> | <u>Boat_name</u> | Weekday |
|----------------|--------------------|------------------|---------|

- Sailor_name کلید اصلی جدول Sailor است.
- Boat_name کلید اصلی جدول Boat است.
- Sailor_name و Boat_name با هم کلید اصلی جدول Reserve هستند و به ترتیب کلید خارجی به جداول Sailor و Boat هستند.
- "فیلد رتبه" برای قایق‌ها نشان دهنده حداقل رتبه‌ای است که یک ملوان برای رزرو آن قایق نیاز دارد.

1 – جدول زیر را ایجاد کنید.

الف) محدودیت‌های زیر را اعمال کنید:

Sailor_name ها از نوع int بوده، identity و از مقدار 300 شروع می‌شوند.

Boat_color از نوع varchar و با حداکثر 30 کاراکتر

Boat_rank ها دارای مقدار حداقل 70 و حداکثر 150 باشند.

ب) برای هر جدول 4 داده پی‌مختلف را وارد کنید.

ج) پرس و جو های زیر را بنویسید:

- نام کلیه قایق‌هایی که در روز شنبه به آب انداخته شده‌اند.
- ایجاد دید (view) که نام کشتی، نام قاپران و رنگ کشتی را نشان دهد.
- نمایش رنگ کلیه قایق‌ها.
- نمایش نام قایق‌هایی که در روزهای شنبه یا دوشنبه به آب انداخته شده‌اند.
- رنگ تمامی قایق‌هایی که در روزی که نام آن با "s" شروع می‌شود، به آب انداخته شده‌اند.

(الف)

```

~vs8743.sql - NYX....er (NYX\Samin (60))
Create Table Sailor(
  Sailor_name int not null primary key,
  Sailor_rank int );

Create Table Boat(
  Boat_name varchar(20)not null primary key,
  Boat_color varchar(30),
  Boat_rank int constraint rank_limit check(Boat_rank<= 150 AND Boat_rank>=70)
);

Create Table Reserve(
  Sailor_name int not null,
  Boat_name varchar(20)not null,
  Weekday varchar(15),
  foreign key (Sailor_name) references Sailor(Sailor_name),
  foreign key (Boat_name) references Boat(Boat_name),
  primary key (Boat_name, Sailor_name)
);
100 %
Messages
Commands completed successfully.

Completion time: 2023-03-04T14:57:42.8409627+03:30

```

(ب)

```

insert into Sailor values (300,1);
insert into Sailor values (301,2);
insert into Sailor values (302,3);
insert into Sailor values (303,4);
.00 %
Messages
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-03-04T15:02:52.2876244+03:30

```

```
insert into Boat values ('A', 'Blue', 70);  
insert into Boat values ('B', 'Red', 80);  
insert into Boat values ('C', 'Orange', 90);  
insert into Boat values ('D', 'Yellow', 100);
```

00 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-03-04T15:04:52.2750107+03:30

```
insert into Reserve values (300, 'A', 'Sat');  
insert into Reserve values (301, 'B', 'Sun');  
insert into Reserve values (302, 'C', 'Mon');  
insert into Reserve values (303, 'D', 'Tue');
```

.00 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-03-04T15:06:47.9143542+03:30

(ج

-۱

```
SELECT Boat_name
FROM Reserve
WHERE Weekday = 'Sat';
```

100 %

Results Messages

| | Boat_name |
|---|-----------|
| 1 | A |

-۲

```
CREATE VIEW [Boat_Sailor_Color] AS
SELECT Boat.Boat_name, Reserve.Sailor_name, Boat.Boat_color
FROM Boat INNER JOIN Reserve ON Boat.Boat_name = Reserve.Boat_name;
```

100 %

Messages

Commands completed successfully.

Completion time: 2023-03-04T15:43:06.9654930+03:30

-۳

```
SELECT Boat_rank
FROM Boat
```

.00 %

Results Messages

| | Boat_rank |
|---|-----------|
| 1 | 70 |
| 2 | 80 |
| 3 | 90 |
| 4 | 100 |

-۴

```
SELECT Boat.Boat_name
FROM Boat INNER JOIN Reserve ON Boat.Boat_name = Reserve.Boat_name
WHERE Reserve.Weekday='Sun' OR Reserve.Weekday='Mon';
```

100 %

Results Messages

| | Boat_name |
|---|-----------|
| 1 | B |
| 2 | C |

-۵

```
SELECT DISTINCT Boat.Boat_color
FROM Boat INNER JOIN Reserve ON Boat.Boat_name = Reserve.Boat_name
WHERE Reserve.Weekday LIKE 'S%';
```

.00 %

Results Messages

| | Boat_color |
|---|------------|
| 1 | Blue |
| 2 | Red |

2- جدول زیر را وارد کنید.

| Emp | | | Dept | | | Prj | | |
|-----------------------|------------------------|--------|-------------|-----------------|--------|-----------------------|----------|--------|
| محدودیت | نوع | ستون | محدودیت | نوع | ستون | محدودیت | نوع | ستون |
| Primary Key, Identity | عدد صحیح | Emp_id | Primary Key | عدد صحیح | DeptNo | Primary Key, Identity | عدد صحیح | Prj_id |
| | رشته حداکثر ۵۰ کاراکتر | Name | | رشته ۵۰ کاراکتر | Name | | | Emp_id |
| Foreign Key | عدد صحیح | DeptNo | | | | | | |

در ادامه جدول هارا آماده میکنیم:

(۱)

```

Create Table Emp(
  Emp_id int not null identity primary key,
  Name varchar(50),
  Salary int,
);

```

100 %

Messages

Commands completed successfully.

Completion time: 2023-03-04T15:59:51.3266503+03:30

(۲)

```

Create Table employee_project(
  prj_id int not null ,
  emp_id int not null ,
  foreign key (prj_id) references Project(id),
  foreign key (emp_id) references Emp(Emp_id),
  primary key (prj_id, emp_id)
);

```

00 %

Messages

Commands completed successfully.

Completion time: 2023-03-04T16:08:09.5658175+03:30

(۳)

```

Create Table Project(
  id int not null identity primary key,
  name varchar(15),
);

```

100 %

Messages

Commands completed successfully.

Completion time: 2023-03-04T16:02:58.3154355+03:30

الف) پروژه های زیر را در جدول وارد کنید:

- (۱) پروژه ای A: کارمند manager (1)، کارمند b (2)، کارمند e (2)
- (۲) پروژه ای B: کارمند c (3)، کارمند d (3)، کارمند a (2)، کارمند b (1)
- (۳) پروژه ای C: کارمند a (1)، کارمند f (2)
- (۴) پروژه ای D: کارمند manager (1)، کارمند f (2)، کارمند h (3)، کارمند i (4)، کارمند g (2)، کارمند k (3)

-۱

```

-- Insert data into the Emp table
INSERT INTO Emp (Name, Salary)
VALUES ('Manager', 100000),
       ('EmpB1', 50000),
       ('EmpB2', 50000),
       ('Emp01', 45000),
       ('Emp02', 45000);

-- Insert data into the Project table
INSERT INTO Project (name)
VALUES ('A');

-- Insert data into the employee_project table
INSERT INTO employee_project (prj_id, emp_id)
VALUES (1, 1), -- Manager assigned to project A
       (1, 2), -- EmpB1 assigned to project A
       (1, 3), -- EmpB1 assigned to project A
       (1, 4), -- Emp01 assigned to project A
       (1, 5); -- Emp02 assigned to project A

```

%

Messages

(5 rows affected)

(1 row affected)

(5 rows affected)

Completion time: 2023-03-04T16:25:05.5974576+03:30

```
-- Insert data into the Emp table
INSERT INTO Emp (Name, Salary)
VALUES ('EmpC1', 30000),
       ('EmpC2', 30000),
       ('EmpC3', 30000),
       ('EmpD1', 20000),
       ('EmpD2', 20000),
       ('EmpD3', 20000),
       ('EmpA1', 60000),
       ('EmpA2', 60000),
       ('EmpE1', 80000),
       ('EmpE2', 80000),
       ('EmpB3', 50000);

-- Insert data into the Project table
INSERT INTO Project (name)
VALUES ('B');

-- Insert data into the employee_project table
INSERT INTO employee_project (prj_id, emp_id)
VALUES (2, 6), -- EmpC1 assigned to project B
       (2, 7), -- EmpC2 assigned to project B
       (2, 8), -- EmpC3 assigned to project B
       (2, 9), -- EmpD1 assigned to project B
       (2, 10), -- EmpD2 assigned to project B
       (2, 11), -- EmpD3 assigned to project B
       (2, 12), -- EmpA1 assigned to project B
       (2, 13), -- EmpA2 assigned to project B
       (2, 14), -- EmpE1 assigned to project B
       (2, 15), -- EmpE2 assigned to project B
       (2, 16); -- EmpB3 assigned to project B
```

100 %

Messages

100 %

✓ Query executed successfully.


```
-- Insert data into the Emp table
INSERT INTO Emp (Name, Salary)
VALUES ('EmpA3', 60000),
       ('EmpF1', 30000),
       ('EmpF2', 30000);

-- Insert data into the Project table
INSERT INTO Project (name)
VALUES ('C');

-- Insert data into the employee_project table
INSERT INTO employee_project (prj_id, emp_id)
VALUES (3, 17), -- EmpA3 assigned to project C
       (3, 18), -- EmpF1 assigned to project C
       (3, 19); -- EmpF2 assigned to project C
```

0 %

Messages

100 %

Query executed successfully.

```
-- Insert data into the Emp table
INSERT INTO Emp (Name, Salary)
VALUES ('EmpF3', 50000),
       ('EmpF4', 50000),
       ('EmpH1', 45000),
       ('EmpH2', 45000),
       ('EmpH3', 45000),
       ('EmpI1', 65000),
       ('EmpI2', 65000),
       ('EmpI3', 65000),
       ('EmpI4', 65000),
       ('EmpG1', 55000),
       ('EmpG2', 55000),
       ('EmpK1', 40000),
       ('EmpK2', 40000),
       ('EmpK3', 40000);

-- Insert data into the Project table
INSERT INTO Project (name)
VALUES ('D');

-- Insert data into the employee_project table
INSERT INTO employee_project (prj_id, emp_id)
VALUES (4, 1), -- Manager assigned to project D
       (4, 20), -- EmpF3 assigned to project D
       (4, 21), -- EmpF4 assigned to project D
       (4, 22), -- EmpH1 assigned to project D
       (4, 23), -- EmpH2 assigned to project D
       (4, 24), -- EmpH3 assigned to project D
       (4, 25), -- EmpI1 assigned to project D
       (4, 26), -- EmpI2 assigned to project D
       (4, 27), -- EmpI3 assigned to project D
       (4, 28), -- EmpI4 assigned to project D
       (4, 29), -- EmpG1 assigned to project D
       (4, 30), -- EmpG2 assigned to project D
       (4, 31), -- EmpK1 assigned to project D
       (4, 32), -- EmpK2 assigned to project D
```

ب) دستورات زیر را اجرا کنید:

(۱) لیستی از پروژه هایی که کمتر از ۴ کارمند در آن ها کار می کنند.

(۲) لیستی از کارمندان به همراه نام بخش مربوطه، نام پروژه ها.

(۳) مجموع حقوق افرادی که در پروژه ی B شرکت دارند.

(۴) لیستی از پروژه ها به همراه حقوق متوس افرادی که در آن شرکت دارند.

(۵) لیست پروژه هایی که manager در آنها دخیل است.

(۱)

```
SELECT Project.name
FROM Project INNER JOIN employee_project ON Project.id = employee_project.prj_id
GROUP BY Project.name
HAVING COUNT(employee_project.emp_id) < 4;
```

100 %

Results Messages

| | name |
|---|------|
| 1 | C |

(۲)

```
SELECT Emp.Name, Project.name AS ProjectName
FROM Emp
JOIN employee_project ON Emp.Emp_id = employee_project.emp_id
JOIN Project ON employee_project.prj_id = Project.id;
```

100 %

Results Messages

| | Name | ProjectName |
|---|---------|-------------|
| 1 | Manager | A |
| 2 | EmpB1 | A |
| 3 | EmpB2 | A |
| 4 | EmpO1 | A |
| 5 | EmpO2 | A |
| 6 | EmpC1 | B |
| 7 | EmpC2 | B |
| 8 | EmpC3 | B |

Query executed successfully.

```

SELECT SUM(Salary)
FROM Emp
WHERE Emp_id IN (
  SELECT emp_id
  FROM employee_project
  WHERE prj_id = (
    SELECT id
    FROM Project
    WHERE name = 'B'
  )
);

```

100 %

Results Messages

| | (No column name) |
|---|------------------|
| 1 | 480000 |

(۳)

```

SELECT p.name AS project_name, AVG(e.salary) AS avg_salary
FROM Project p
INNER JOIN employee_project ep ON p.id = ep.prj_id
INNER JOIN Emp e ON ep.emp_id = e.Emp_id
GROUP BY p.name

```

100 %

Results Messages

| | project_name | avg_salary |
|---|--------------|------------|
| 1 | A | 58000 |
| 2 | B | 43636 |
| 3 | C | 40000 |
| 4 | D | 55000 |

(۴)

```

SELECT Project.name
FROM Project
INNER JOIN employee_project ON Project.id = employee_project.prj_id
INNER JOIN Emp ON employee_project.emp_id = Emp.Emp_id
WHERE Emp.Name = 'Manager';

```

(۵)

کد آزمایش پیوست شده است.