

تمرین ۲ پایگاه داده  
حساب آگهی دستی است

A (AID, A1)

B (AID, BID, B1)

C (AID, BID, C1)

D (AID, BID, D1)

E (EID, E1)

R<sub>r</sub> (AID, BID, EID, X)

طیله اصلی: مسکنی که به طور مشخص به فرد یک ردیف یا رکورد را در یک رابطه مشخص می کند.  
طیله خارجی: طیله داخلی جدول والد است که در جدول مورد استفاده قرار می گیرد و بین جدول والد و فرزند ارتباط برقرار می کند.  
طیله اصلی مقدار مشخص به فرد جدول است و طیله خارجی مقدار مشخص به فرد جدول والد است که در جدول استفاده می شود.  
(این ۲ جدول را محدود می کند.)

T<sub>1</sub> (A, B, C, FK)

T<sub>2</sub> (D, E, W, FK)

T<sub>3</sub> (FK, X, Y, Z)

(update T<sub>2</sub> در T<sub>3</sub> در رابطه)

update T<sub>3</sub>  
Set FK=18  
where FK=10

?

آلده FK=10 در جدول T<sub>2</sub> با FK=18 چون FK=10 است و FK=18 در جدول T<sub>3</sub> قرار می گیرد.  
مقدار در جدول ها تغییر نمی کند.

آلده FK=10 در جدول T<sub>2</sub> با FK=18 در جدول T<sub>1</sub> که FK=18 است مقدار آن به 18 تغییر می کند.

Person (SSN, DriverLicenseNo, Name, Address, OID)

BANK (BName, BAAddress, OID)

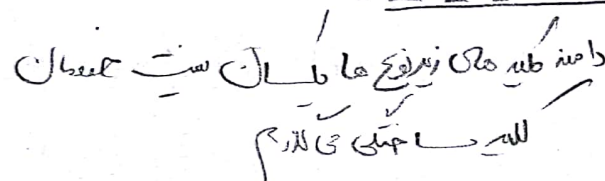
COMPANY (CName, CAddress, OID)

CAR (VehicleId, Cstyle, Cmake, CModel, CYear, LicensePlateNo)

TRUCK (VehicleId, Tonnage, TModel, TYear, TMake, LicensePlateNo)

REGISTERED-VEHICLE (LicensePlateNo, VID)

OWNS (LoanOrRegular, PurchaseDate, LicensePlateNo, OID, VID)

OWNER (OID) → 

S (SID, SNAME, SDEG, SMJR, SDEID)  
C (CID, CTITLE, CREDIT, CTYPE, CDEID)  
SC (SID, CID, TR, GRADE)

1) select CID from SC where SID=123

2) select CTITLE from C except

select C.TITLE from SC

Inner join C on C.CID = SC.CID

where SID=123

2) select S.SID, C.CID from S,C except  
select SC.SID, SC.CID from SC

✓) update SC set GRADE=20  
where SID=123 and CID =  
(select CID from C  
where CTITLE = 'DataBase')

1)

دراسة ایاہ اضافہ شود

insert into C (CID) values (500);

insert into SC values (123, 500, 0001, 16)

- DESC =>

م صورت نزولی مرتبگی کند

- GRANT =>

امتیازات و نقش هایی را به حساب ها و نقش های کاربری اختصاص می دهد

- REVOKE =>

به مدیران دسترسی امکان می دهد تا امتیازات و نقش هایی را از افراد و نقش ها بگیرند (برای مثال نقش را از کاربر بگیرد)

- HAVING =>

به جای دستور WHERE در توابع کل استفاده می شود (aggregate function)

- COUNT(\*) =>

تعداد رکورد های جدول را می گویند (تعداد سطرها)

- COUNT(1) =>

طایفه دستور COUNT(+) است اما کارایی کمتر نیاز به دانستن هیچ چیزی ندارد و فقط مقدار 1 را باز می می کند

- GROUP BY

در conjunction با توابع کل استفاده می شود تا مجموع نتایج توابع را به دست آورد

- LIKE

در WHERE استفاده می شود تا به الگوی در ستون ارجحیت دهد و به الی

- UNION

ترکیب نتیجه 2 یا بسته از عبارت SELECT استفاده می شود

- UNION ALL

با استفاده از بوند ALL می توان همه موارد حتی موارد تکراری را اضافه کرد

ل در انتهای Pdf مشاهده است

1) Select SNAME from S where S# in (  
select SP.S# from SP  
where SP.P# = 'P2');

2) select SNAME from S where EXISTS(  
select 1 from SP  
where SP.P# = 'P2' AND SP.S# = S.S#);

3) select distinct SNAME from S where S# in (  
select SP.S# from SP where SP.P# in (  
select P.P# from P where P.color = 'RED');

4) select <sup>P#</sup>YMAX(GTY) from SP group by P#;

5) select P# from SP join P  
where SP.S# = 'S2' or P.WEIGHT > 16;

6) update SP set QTY = 900 where QTY = 500

7) Delete from S where S# in (  
select SP.S# from SP where SP.P# in (  
select P.P# from P where P.color = 'BLUE');

۹ برای اجرای این دستور باید یک مقدار را به جدول داده شود که در مقادیر بالا به کار برده شده است  
 با استفاده از تعریف ایندکس می‌توانیم به راحتی در جدول موجودیت را می‌توانیم در یک بخش خاص  
 جستجو کنیم (برای تعریف ایندکس باید مقدار اول مقدار غیر null باشد) ایندکس بسیار کمتری است که می‌تواند ایندکس را  
 مرتب می‌کند و به ایندکس از جدول گرفته می‌شود و اگر در جدول رابطه در جدول اصلی داده‌ها است را در جدول ایندکس کار  
 می‌کند و ایندکس می‌کند اما در مواردی که مقدار داده‌ها زیاد است و می‌تواند زمان کار را بسیار  
 کم کند.

CREATE INDEX index1 ON student-DB(SID)

ON student-DB(SID)

SELECT \* FROM student-DB

WHERE SID = 14000702



A	B	C	F	B
a <sub>1</sub>	b <sub>1</sub>	Mk	f <sub>1</sub>	b <sub>1</sub>

select \*

From y right outer join x

on y.B = x.B

where y.C like 'M%'

نتیجہ کار  
نتیجہ کار

نتیجہ کار =

7) در ابتدا دو جدول S و C را میسازیم و مقادیر گفته شده را اضافه میکنیم.

```
CREATE TABLE S (  
    ID INTEGER PRIMARY KEY,  
    NAME TEXT  
);  
  
INSERT INTO S VALUES (1, 'ALI');  
INSERT INTO S VALUES (2, 'REZA');  
INSERT INTO S VALUES (3, 'AMIR');  
  
CREATE TABLE C (  
    ID INTEGER,  
    C_NAME TEXT,  
    FOREIGN KEY (ID) REFERENCES S (ID)  
);  
  
INSERT INTO C VALUES (1, 'DATABASE');  
INSERT INTO C VALUES (1, 'NETWORK');  
INSERT INTO C VALUES (2, 'DATABASE');  
INSERT INTO C VALUES (2, 'NETWORK SECURITY');
```

جداول S و C به شکل زیر بدست می آیند.

	ID	NAME
1	1	ALI
2	2	REZA
3	3	AMIR

جدول S

	ID	C_NAME
1	1	DATABASE
2	1	NETWORK
3	2	DATABASE
4	2	NETWORK SECURITY

جدول C

ایدی علی را به 5 تغییر میدهم.

```
UPDATE S SET ID = 5 WHERE ID = 1;
```

	ID	NAME
1	2	REZA
2	3	AMIR
3	5	ALI

سطر رضا را پاک میکنیم.

```
DELETE FROM S WHERE ID = 2;
```

	ID	NAME
1	3	AMIR
2	5	ALI

یک جدول جدید با نام STUDENT از S میسازیم و S را پاک میکنیم.

```
CREATE TABLE STUDENT AS SELECT * FROM S;  
DROP TABLE S;
```

	ID	NAME
1	3	AMIR
2	5	ALI

مقادیر داده شده را همزمان به STUDENT اضافه میکنیم.

```
INSERT INTO STUDENT(ID, NAME) VALUES  
  (4, 'Alireza'),  
  (5, 'Amirali'),  
  (6, 'Mohamad');
```

	ID	NAME
1	3	AMIR
2	5	ALI
3	4	Alireza
4	5	Amirali
5	6	Mohamad

```
CREATE TABLE S (  
    ID INTEGER PRIMARY KEY,  
    NAME TEXT  
);  
  
INSERT INTO S VALUES (1, 'ALI');  
INSERT INTO S VALUES (2, 'REZA');  
INSERT INTO S VALUES (3, 'AMIR');  
  
CREATE TABLE C (  
    ID INTEGER,  
    C_NAME TEXT,  
    FOREIGN KEY (ID) REFERENCES S (ID)  
);  
  
INSERT INTO C VALUES (1, 'DATABASE');  
INSERT INTO C VALUES (1, 'NETWORK');  
INSERT INTO C VALUES (2, 'DATABASE');  
INSERT INTO C VALUES (2, 'NETWORK SECURITY');  
  
UPDATE S SET ID = 5 WHERE ID = 1;  
  
DELETE FROM S WHERE ID = 2;  
  
CREATE TABLE STUDENT AS SELECT * FROM S;  
DROP TABLE S;  
  
INSERT INTO STUDENT (ID, NAME) VALUES  
    (4, 'Alireza'),  
    (5, 'Amirali'),  
    (6, 'Mohamad');
```