```
INSERT INTO tableName (attributes) VALUES (values);
INSERT INTO Likes (beer, drinker) VALUES ('Bud', 'Sally');
CREATE TABLE Likes (
    beer      CHAR(30),
    drinker   CHAR(20)  DEFAULT  'anyName'
);
```

```
CREATE TABLE Drinkers (
    name    CHAR(30)  PRIMARY KEY,
    addr    CHAR(50)  DEFAULT  '123 Main St.',
    phone   CHAR(16)  DEFAULT  NULL
);
```

```
DELETE FROM table WHERE Conditions;
DELETE FROM Drinkers
WHERE name = 'Michael';
```

```
INSERT INTO PotBuddies          -- creating new table Pot Buddies
(SELECT ~~~                     -- insert into it the values from others
  FROM ~~~
  WHERE ~~~
);
```

```
-- Delete whole table and records
    DELETE FROM Likes;
```

* <u>Foreign Key :</u>

- <u>As an attribute :</u>

```
CREATE TABLE Beers (
    name    CHAR(20)   PRIMARY KEY ,
    manf    CHAR (20) );

CREATE TABLE Sells (
    bar     CHAR(20) ,
    beer    CHAR (20)  REFERENCES  Beers ( name ) ,
    price   REAL );
```

- <u>As schema element :</u>

```
CREATE TABLE Sells (
    bar     CHAR(20),
    beer    CHAR (20),
    price   REAL ,
    FOREIGN KEY (beer) REFERENCES  Beers(name)
            ON DELETE  SET NULL
            ON UPDATE  CASCADE );
ALTER TABLE Beers
      ADD FOREIGN KEY (name) REFERENCES  Sells(beers);
ALTER TABLE Product
      ADD PRIMARY KEY (model);
```

```
UPDATE   Sells
SET      price = 4.00
WHERE    price > 4.00;
```

```
UPDATE   Drinkers
SET      phone = '111-222-3333'
WHERE    name = 'Fred';
```

```
UPDATE   Sells
SET price = CASE
              WHEN  price < 4.00
                 THEN price = price * 1.1
             WHEN  price = 4.0
                THEN price = 4.12
             ELSE
                price = price * 1.05
        END;
```

SQL
- Only one
- more than
- topmost value

Sells laptops, but donot sell PC

```
SELECT DISTINCT maker
FROM    product
WHERE   type = 'laptop'
EXCEPT
SELECT DISTINCT maker
FROM    product
WHERE   type = 'pc';
```

find the laptops with speed slower than any PC

```
SELECT  model
FROM    laptop
WHERE   speed <= ANY (SELECT  speed
                      FROM    pc);
```

find the printers with the highest price

```
SELECT   model
FROM     printer
WHERE    price  >= ALL(SELECT price  FROM  printer);
 WHERE   price  = (SELECT  MAX(price)  FROM  printer);
```

```
CREATE TABLE  Sells (
    bar     CHAR(20),
    beer    CHAR(20),
    price   REAL,
    FOREIGN KEY (beer) REFERENCES  Beers (name)
        ON DELETE  SET  NULL
        ON UPDATE  CASCADE
);
```

Exercise 4.6.1

a) Straight E-R method:

Depts(name, chair)

Courses(courseNumber, name, room)

LabCourses(courseNumber, DeptsName, allocation)

b) Object-oriented method:

Depts(name, chair)

Courses(courseNumber, DeptsName, room)

LabCourses(courseNumber, DeptsName, room, allocation)

c) Null method

Depts(name, chair)

Courses(courseNumber, DeptsName, room, allocation)

Index:

```
CREATE INDEX manfInd ON Beers(manf);
CREATE INDEX sellsInd ON Sells(bar, beer);
```

```
CREATE VIEW viewName AS SELECT
                        FROM
                        WHERE  ____________  ;
```

```
ALTER TABLE product
      ADD PRIMARY KEY (model);
ALTER TABLE pc
      ADD FOREIGN KEY (model) REFERENCES product(model);
ALTER TABLE laptop
      ADD FOREIGN KEY (model) REFERENCES product(model);
```

Average : AVG   rename : AS

GROUP BY : ~ keyword each, every,

For each manufacturer, the avg screen size its laptops

```
SELECT   maker, AVG(screen)
FROM     product p, laptop l
WHERE    p.model = l.model
GROUP BY maker;
```

```
SELECT   maker, AVG(screen)
FROM     product p, laptop l
WHERE    p.model IN
           (SELECT model
            FROM l.model)
GROUP BY maker;
```

Find the manufacturers that make at least three diff. models of PC

```
SELECT   maker
FROM     product
WHERE    model IN (SELECT model
                   FROM pc)
GROUP BY maker
HAVING COUNT (maker) >= 3;
```

```
SELECT   maker
FROM     product
WHERE    type = 'pc'
GROUP BY maker
HAVING COUNT (maker) >= 3;
```

```
SELECT   maker , MAX(pc.price)
FROM
```

```
UPDATE table
SET    attributes
WHERE  conditions;
```

```
DELETE FROM table
WHERE  conditions;
```

```
INSERT INTO table
   VALUES (    );
```

```
DELETE FROM laptop
WHERE  model  IN  (SELECT model
                   FROM product
                   WHERE type = 'printer');
```