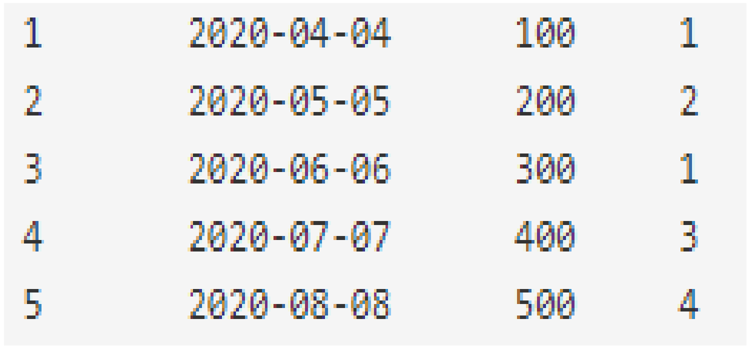
Information Management System Lab

ECSE211L

Bennett University

Orders table

```
create table orders(
oid int(10),
orderdate date,
amount varchar(20),
cid int(10));
```

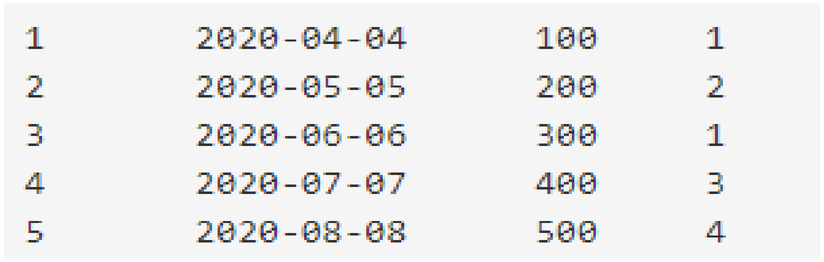


```
1     2020-04-04    100    1
2     2020-05-05    200    2
3     2020-06-06    300    1
4     2020-07-07    400    3
5     2020-08-08    500    4
```

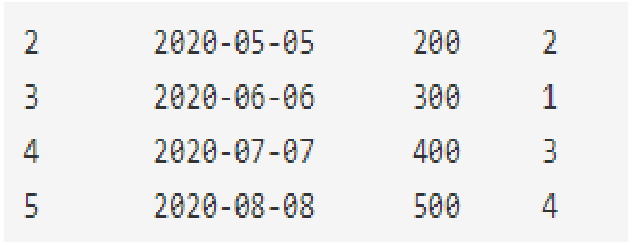
```
INSERT INTO orders (oid, orderdate,amount,cid) values ('1','2020-04-04','100','1');
INSERT INTO orders (oid, orderdate,amount,cid) values ('2','2020-05-05','200','2');
INSERT INTO orders (oid, orderdate,amount,cid) values ('3','2020-06-06','300','1');
INSERT INTO orders (oid, orderdate,amount,cid) values ('4','2020-07-07','400','3');
INSERT INTO orders (oid, orderdate,amount,cid) values ('5','2020-08-08','500','4');

select * from orders;
```

Between Clause

select * from orders where amount >=200 and amount <=500;

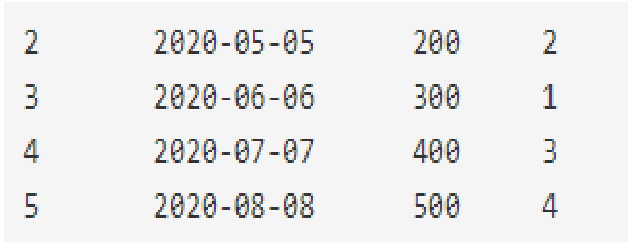


oid	orderdate	amount	cid
1	2020-04-04	100	1
2	2020-05-05	200	2
3	2020-06-06	300	1
4	2020-07-07	400	3
5	2020-08-08	500	4



2	2020-05-05	200	2
3	2020-06-06	300	1
4	2020-07-07	400	3
5	2020-08-08	500	4

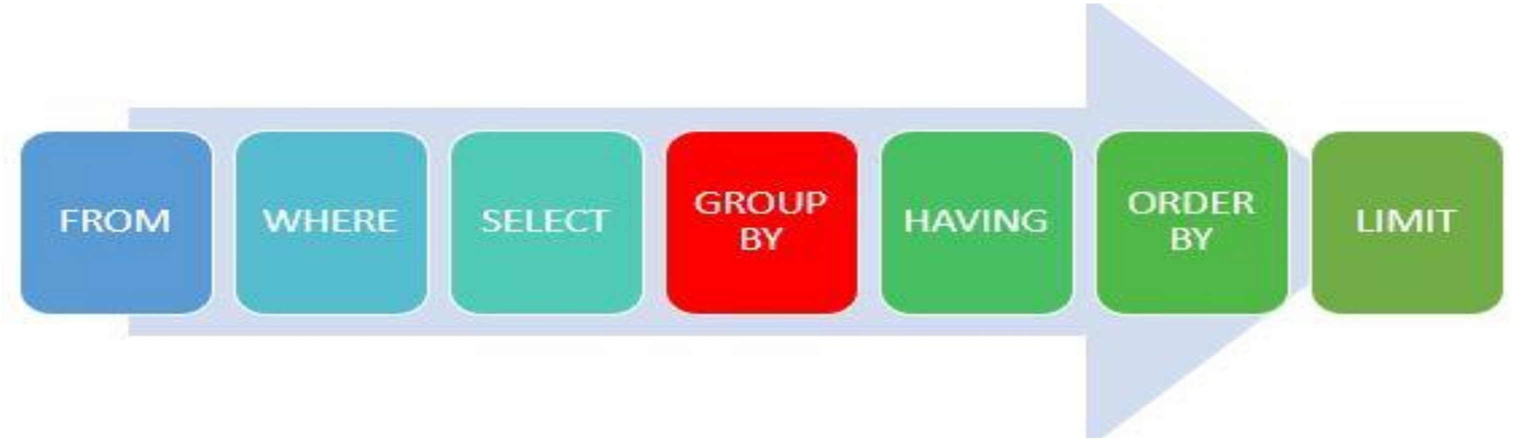
select * from orders where amount BETWEEN 200 and 500;



2	2020-05-05	200	2
3	2020-06-06	300	1
4	2020-07-07	400	3
5	2020-08-08	500	4

Group By Clause

- It groups a set of rows by values of columns or expressions.
- It returns one row for each group.
- It is used with aggregate functions such as SUM, AVG, MAX, MIN and COUNT.



Group By Clause Contd…

Syntax:

```
Select c1, c2 …..
From Tablename
where condition
Group By c1, c2, …, cn;
```

Example:

```
SELECT cid, count(*)
FROM orders
GROUP BY cid;
```

HAVING Clause

- It is added to SQL because the WHERE keyword could not be used with aggregate functions.

Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

Example

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5
ORDER BY COUNT(CustomerID) DESC;
```

Count, Group By, IN, MAX, MIN Clause

oid	orderdate	amount	cid
1	2020-04-04	100	1
2	2020-05-05	200	2
3	2020-06-06	300	1
4	2020-07-07	400	3
5	2020-08-08	500	4

SELECT COUNT(*) FROM orders;

```
5
```

SELECT cid, COUNT(*) FROM orders GROUP BY cid;

cid	count
1	2
2	1
3	1
4	1

SELECT oid, amount, cid FROM orders where amount IN(100,300,500);

oid	amount	cid
1	100	1
3	300	1
5	500	4

SELECT MAX(amount) FROM orders;

```
500
```

SELECT MIN(amount) FROM orders;

```
100
```

SUM, AVG, RAND, SQRT Clause

oid	orderdate	amount	cid
1	2020-04-04	100	1
2	2020-05-05	200	2
3	2020-06-06	300	1
4	2020-07-07	400	3
5	2020-08-08	500	4

SELECT SUM(amount) FROM orders;

```
1500
```

SELECT AVG(amount) FROM orders;

```
300.0000
```

SELECT RAND( ), RAND( ), RAND( );

```
0.8548265787352124    0.11729398473845234    0.02199021822026951
```

select SQRT(16);

```
4
```

SQL Aliases

- SQL aliases are used to give a table, or a column in a table, a temporary name.
- Aliases are often used to make column names more readable.
- An alias only exists for the duration of the query.

Syntax

```
SELECT column_name AS alias_name
FROM table_name;
```

Example:

```
SELECT CONCAT(City, " ", Country) AS Address FROM Customers;
```

Advantages:

- There are more than one table involved in a query
- Functions are used in the query
- Column names are big or not very readable
- Two or more columns are combined together

LIKE clause

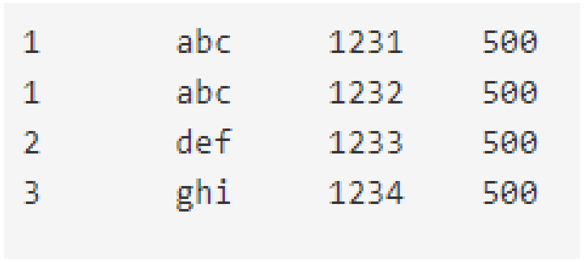
- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.
- There are two wildcards used in conjunction with the LIKE operator:
  - % - The percent sign represents zero, one, or multiple characters
  - _ - The underscore represents a single character
- LIKE Syntax •

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%_%'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"

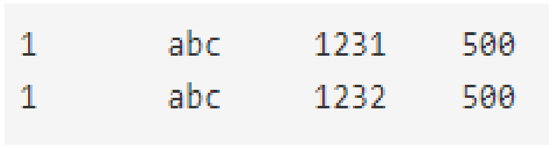
LIKE clause contd…

```
create table orders(
cid int(10),
cust_name varchar (10),
oid int(10),
amount int(20));
```



```
1        abc     1231     500
1        abc     1232     500
2        def     1233     500
3        ghi     1234     500
```

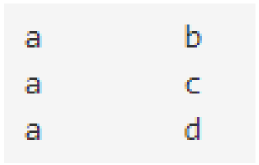
```
INSERT INTO orders (cid, cust_name, oid, amount) values (1, 'abc', 1231, 500);
INSERT INTO orders (cid, cust_name, oid, amount) values (1, 'abc', 1232, 500);
INSERT INTO orders (cid, cust_name, oid, amount) values (2, 'def', 1233, 500);
INSERT INTO orders (cid, cust_name, oid, amount) values (3, 'ghi', 1234, 500);

select * from orders;

select * from orders where cust_name LIKE ‘b%';
```



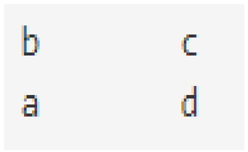
```
1        abc     1231     500
1        abc     1232     500
```

UNION

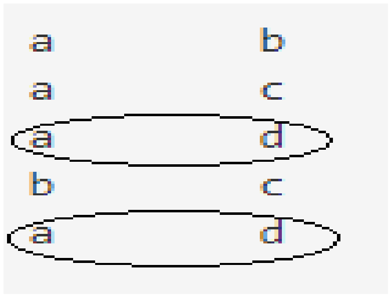
```
create table A(
one varchar(5),
two varchar(5));

create table B(
three varchar(5),
four varchar(5));

INSERT INTO A (one, two) values ('a','b');
INSERT INTO A (one, two) values ('a','c');
INSERT INTO A (one, two) values ('a','d');
INSERT INTO B (three, four) values ('b','c');
INSERT INTO B (three, four) values ('a','d');

SELECT * from A; SELECT * from B;

SELECT one, two FROM A UNION ALL SELECT three, four  FROM B

SELECT one, two FROM A UNION SELECT three, four  FROM B;
```



a	b
a	c
a	d

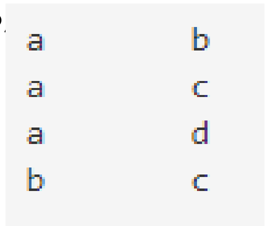


b	c
a	d



a	b
a	c
a	d
b	c
a	d

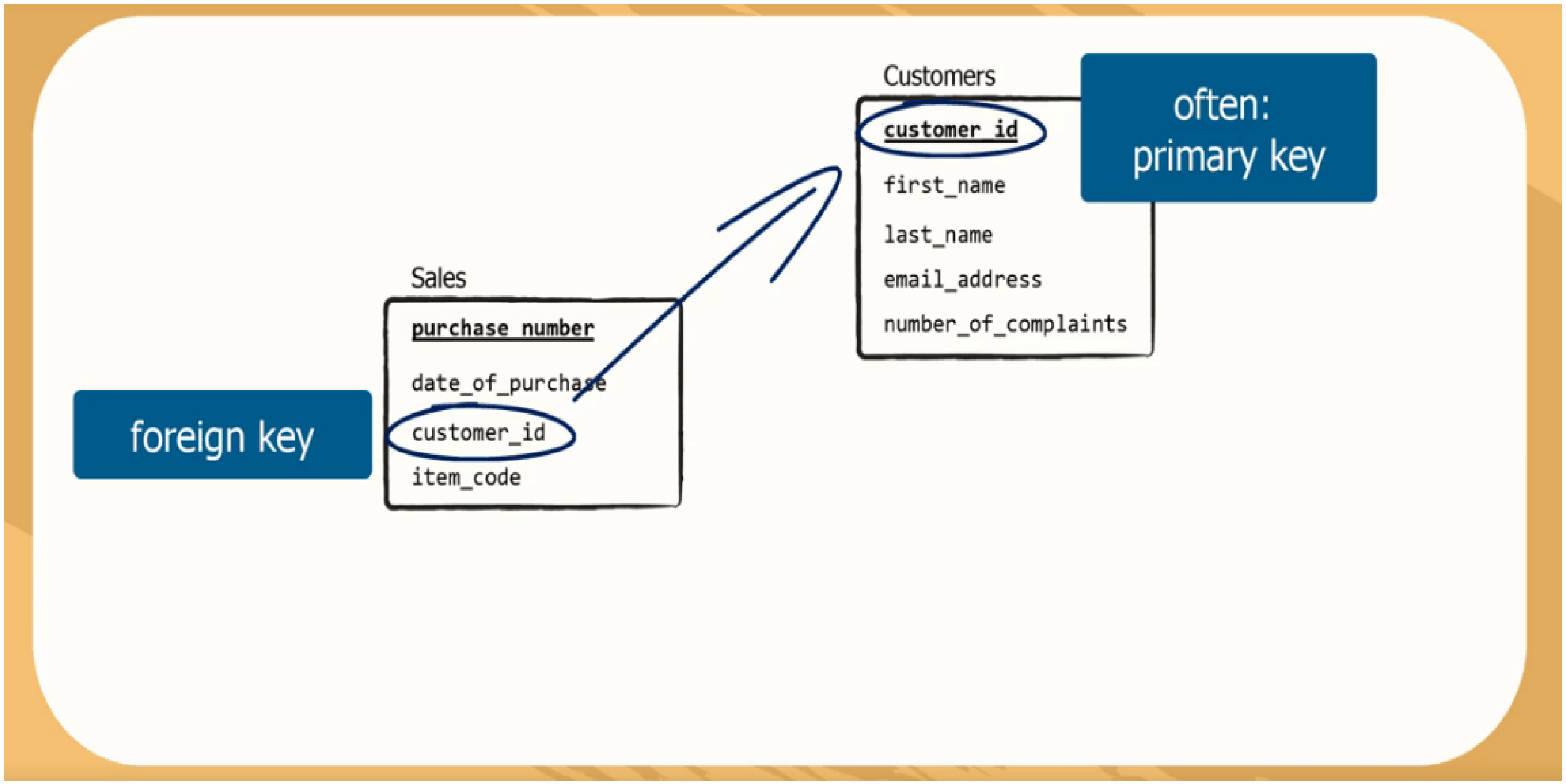
Duplicate Rows



a	b
a	c
a	d
b	c

No Duplicate Rows

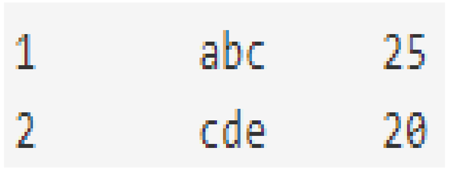
Foreign Key


Customers
customer_id
first_name
last_name
email_address
number_of_complaints
often: primary key
Sales
purchase_number
date_of_purchase
customer_id
item_code
foreign key

SQL FOREIGN KEY Constraint

```
CREATE TABLE student (
   stud_id int AUTO_INCREMENT,
   name VARCHAR(30) NOT NULL,
   age int NOT NULL,
   PRIMARY KEY (stud_id)
);
```

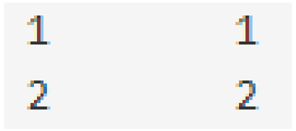
```
insert into student (name, age) values ('abc',25);
insert into student (name, age) values ('cde',20);
select *from student;
```



```
1        abc      25
2        cde      20
```

```
CREATE TABLE enrol(
roll_no int NOT NULL AUTO_INCREMENT,
student_id int NOT NULL,
PRIMARY KEY(roll_no),
FOREIGN KEY (student_id) REFERENCES student
(stud_id)
);
```

```
insert into enrol (stud_id) values (1);
insert into enrol (stud_id) values (2);
select *from enrol;
```



```
1        1
2        2
```

SQL FOREIGN KEY Constraint

```
CREATE TABLE Persons (
   Person_id int AUTO_INCREMENT,
   name VARCHAR(30) NOT NULL,
   age int NOT NULL,
   PRIMARY KEY (Person_id)
);

CREATE TABLE Orders (
   OrderID int NOT NULL,
   OrderNumber int NOT NULL,
   PersonID int,
   PRIMARY KEY (OrderID),
   CONSTRAINT FK_PersonOrder FOREIGN KEY (PersonID)
   REFERENCES Persons(PersonID)
);
```

<u>Add Foreign Key:</u>

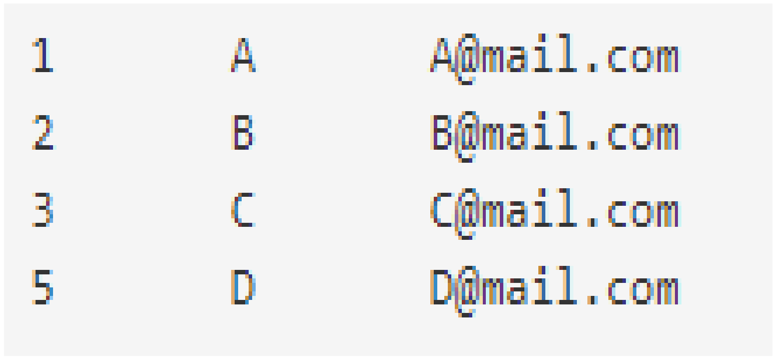
```
ALTER TABLE Orders
ADD CONSTRAINT FK_PersonOrde
r
FOREIGN KEY (PersonID) REFEREN
CES Persons(Person_id);
```

<u>Drop Foreign Key:</u>

```
ALTER TABLE Orders
DROP FOREIGN KEY FK_PersonOr
der;
```

Customers Table

```
Create table customers(
cid int(10),
cname varchar(30),
cemail varchar(20));
```

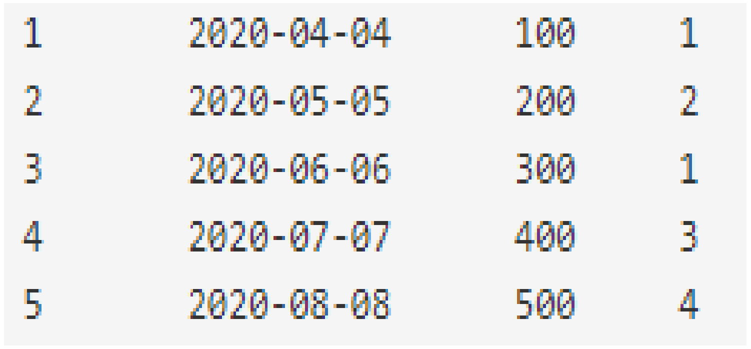


```
1    A    A@mail.com
2    B    B@mail.com
3    C    C@mail.com
5    D    D@mail.com
```

```
INSERT INTO customers (cid,cname,cemail) values ('1','A','A@mail.com');
INSERT INTO customers (cid,cname,cemail) values ('2','B','B@mail.com');
INSERT INTO customers (cid,cname,cemail) values ('3','C','C@mail.com');
INSERT INTO customers (cid,cname,cemail) values ('5','D','D@mail.com');
select * from customers;
```

Orders table

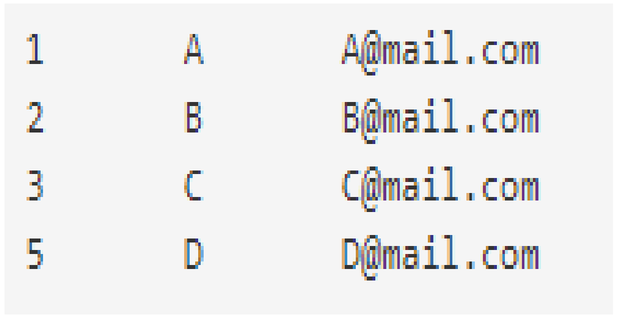
```
create table orders(
oid int(10),
orderdate date,
amount varchar(20),
cid int(10));
```



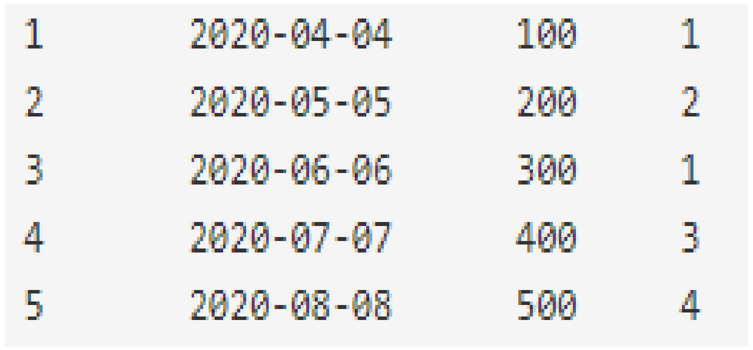
```
1    2020-04-04    100    1
2    2020-05-05    200    2
3    2020-06-06    300    1
4    2020-07-07    400    3
5    2020-08-08    500    4
```

```
INSERT INTO orders (oid, orderdate,amount,cid) values ('1','2020-04-04','100','1');
INSERT INTO orders (oid, orderdate,amount,cid) values ('2','2020-05-05','200','2');
INSERT INTO orders (oid, orderdate,amount,cid) values ('3','2020-06-06','300','1');
INSERT INTO orders (oid, orderdate,amount,cid) values ('4','2020-07-07','400','3');
INSERT INTO orders (oid, orderdate,amount,cid) values ('5','2020-08-08','500','4');

select * from orders;
```

Two Tables

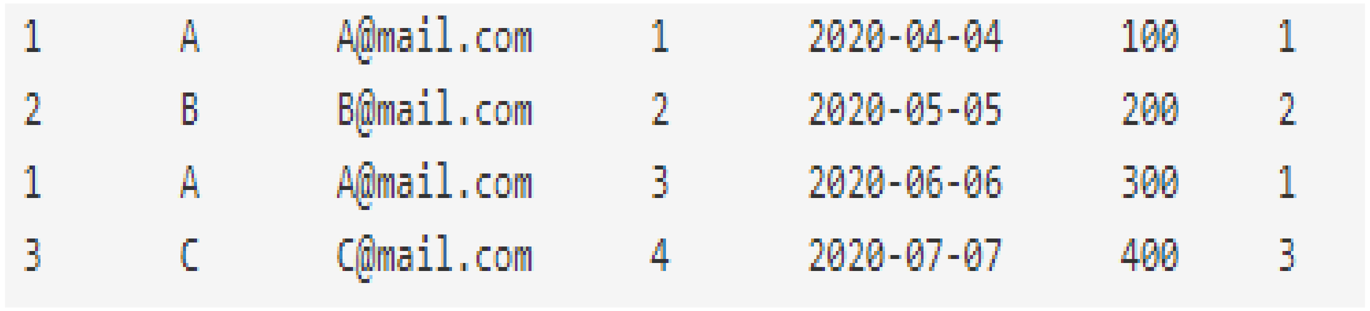


cid	cname	cemail
1	A	A@mail.com
2	B	B@mail.com
3	C	C@mail.com
5	D	D@mail.com



oid	orderdate	amount	cid
1	2020-04-04	100	1
2	2020-05-05	200	2
3	2020-06-06	300	1
4	2020-07-07	400	3
5	2020-08-08	500	4

select * from customers, orders where customers.cid=orders.cid ;



1	A	A@mail.com	1	2020-04-04	100	1
2	B	B@mail.com	2	2020-05-05	200	2
1	A	A@mail.com	3	2020-06-06	300	1
3	C	C@mail.com	4	2020-07-07	400	3