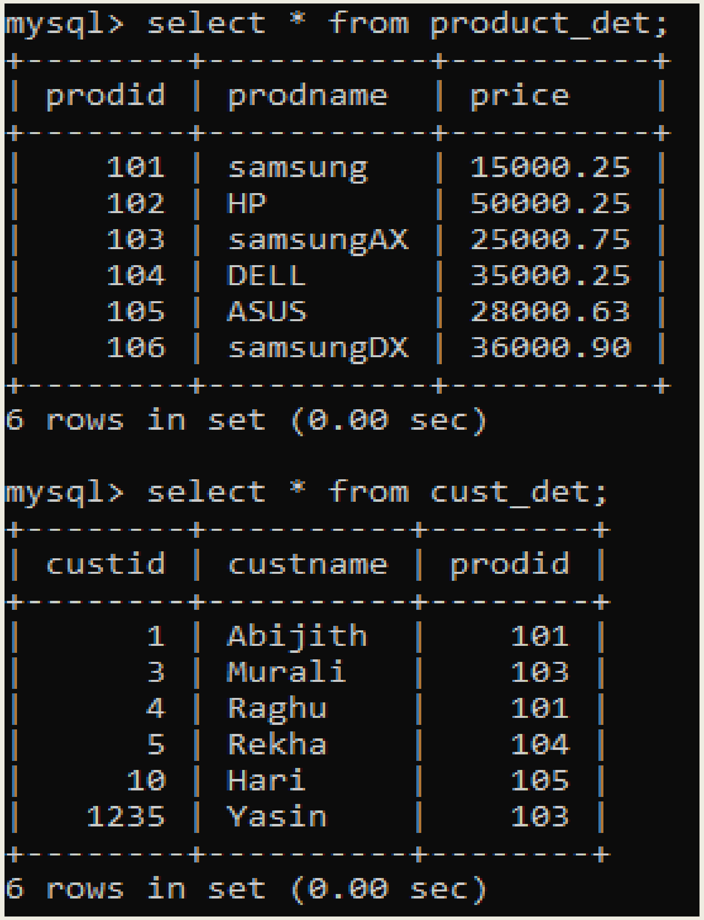
Welcome you all
MySQL
Day 4

D.Sakthivel
Assistant Professor & Trainer,
KGiSL Micro College
KGiSL Campus, Coimbatore – 641 035.

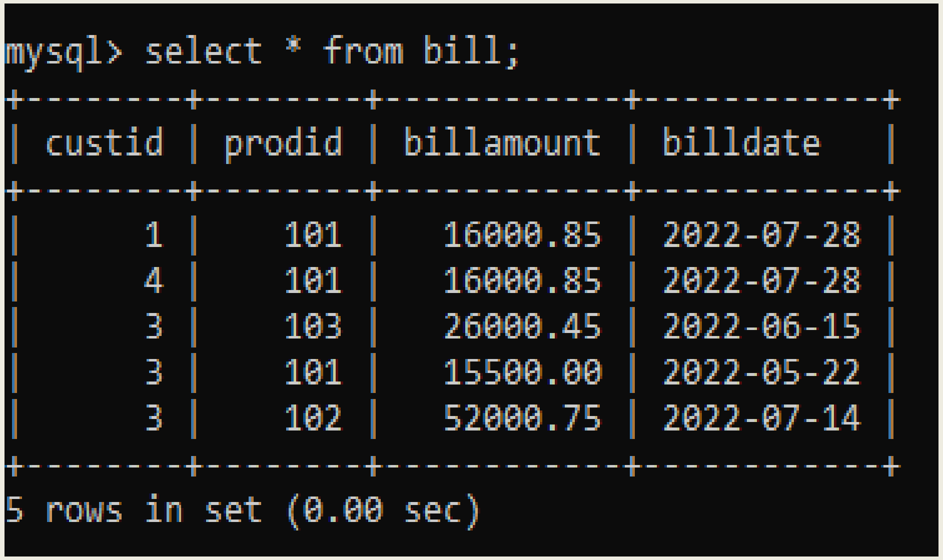
Day 4

- **SUB QUERIES**
- **TRANSACTIONS**
    - **ROLLBACK**
    - **COMMIT**
    - **ACID PROPERTY**
- **VIEWS**

SUB QUERIES



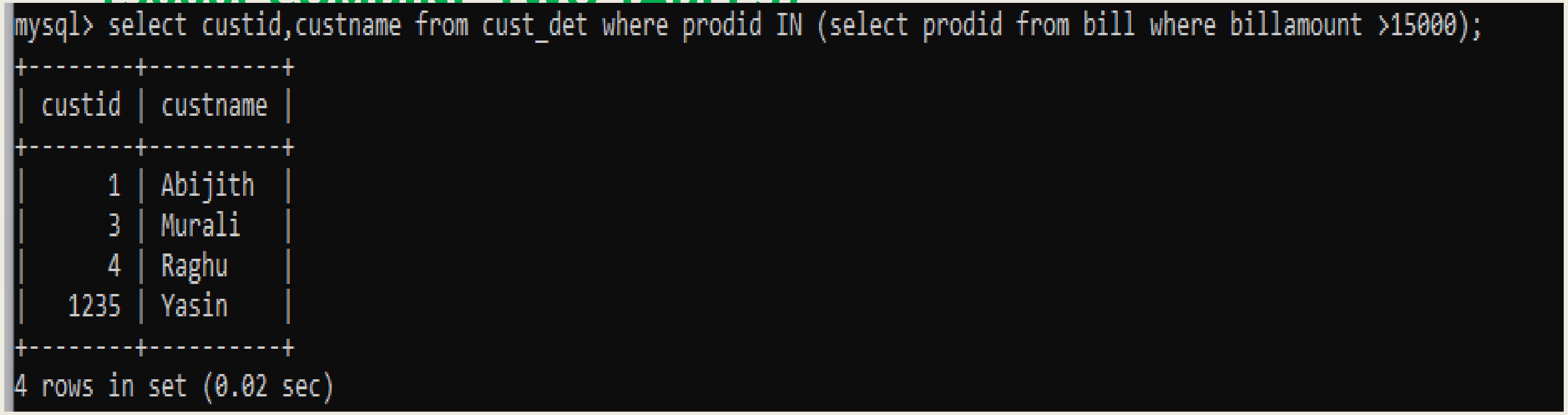
```
mysql> select * from product_det;
+--------+-----------+----------+
| prodid | prodname  | price    |
+--------+-----------+----------+
|    101 | samsung   | 15000.25 |
|    102 | HP        | 50000.25 |
|    103 | samsungAX | 25000.75 |
|    104 | DELL      | 35000.25 |
|    105 | ASUS      | 28000.63 |
|    106 | samsungDX | 36000.90 |
+--------+-----------+----------+
6 rows in set (0.00 sec)

mysql> select * from cust_det;
+--------+----------+--------+
| custid | custname | prodid |
+--------+----------+--------+
|      1 | Abijith  |    101 |
|      3 | Murali   |    103 |
|      4 | Raghu    |    101 |
|      5 | Rekha    |    104 |
|     10 | Hari     |    105 |
|   1235 | Yasin    |    103 |
+--------+----------+--------+
6 rows in set (0.00 sec)
```



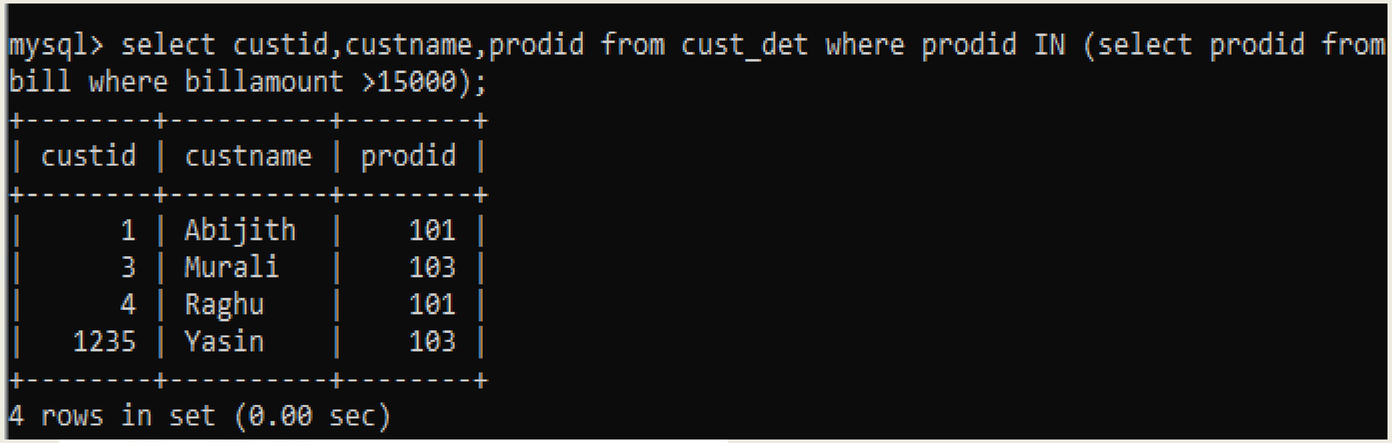
```
mysql> select * from bill;
+--------+--------+------------+------------+
| custid | prodid | billamount | billdate   |
+--------+--------+------------+------------+
|      1 |    101 |   16000.85 | 2022-07-28 |
|      4 |    101 |   16000.85 | 2022-07-28 |
|      3 |    103 |   26000.45 | 2022-06-15 |
|      3 |    101 |   15500.00 | 2022-05-22 |
|      3 |    102 |   52000.75 | 2022-07-14 |
+--------+--------+------------+------------+
5 rows in set (0.00 sec)
```

SUB QUERIES

- TO FIND CUSTOMER WHOSE BILL AMOUNT > 15000(COMBINE TWO TABLES)



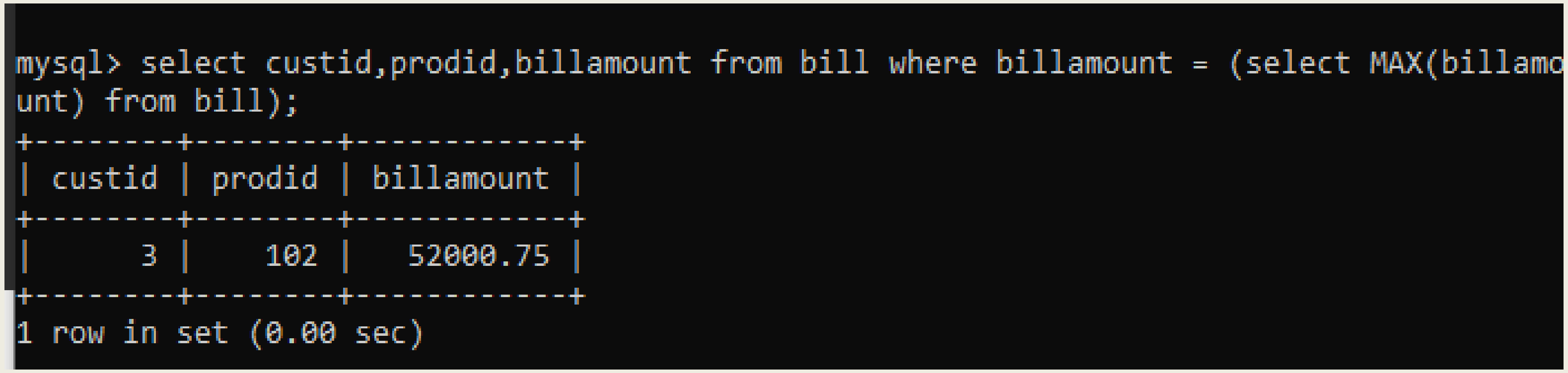
```
mysql> select custid,custname from cust_det where prodid IN (select prodid from bill where billamount >15000);
+--------+----------+
| custid | custname |
+--------+----------+
|      1 | Abijith  |
|      3 | Murali   |
|      4 | Raghu    |
|   1235 | Yasin    |
+--------+----------+
4 rows in set (0.02 sec)
```



```
mysql> select custid,custname,prodid from cust_det where prodid IN (select prodid from
bill where billamount >15000);
+--------+----------+--------+
| custid | custname | prodid |
+--------+----------+--------+
|      1 | Abijith  |    101 |
|      3 | Murali   |    103 |
|      4 | Raghu    |    101 |
|   1235 | Yasin    |    103 |
+--------+----------+--------+
4 rows in set (0.00 sec)
```

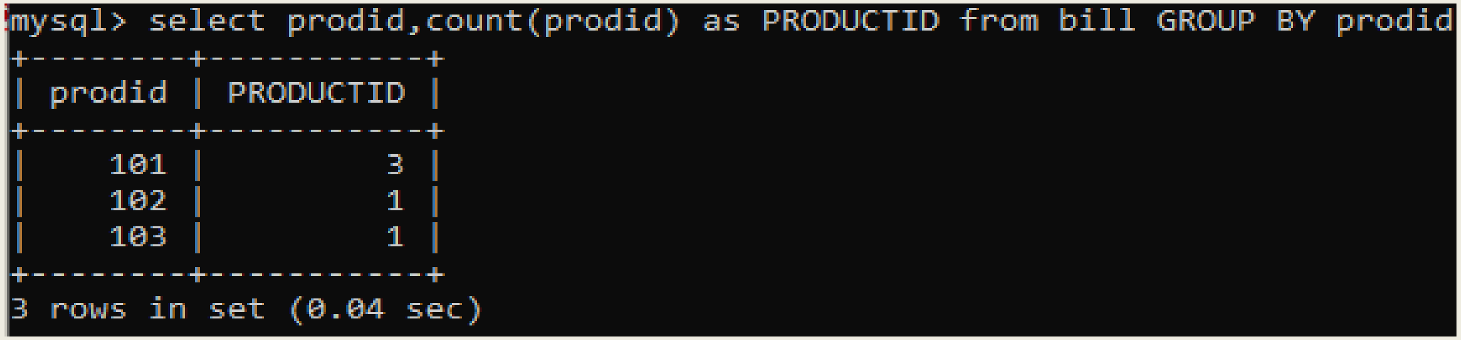
SUB QUERIES

- **TO FIND CUSTOMER OF MAXIMUM BILL AMOUNT .**



```
mysql> select custid,prodid,billamount from bill where billamount = (select MAX(billamo
unt) from bill);
+--------+--------+------------+
| custid | prodid | billamount |
+--------+--------+------------+
|      3 |    102 |   52000.75 |
+--------+--------+------------+
1 row in set (0.00 sec)
```

MySQL correlated subquery



```
mysql> select prodid,count(prodid) as PRODUCTID from bill GROUP BY prodid
+--------+-----------+
| prodid | PRODUCTID |
+--------+-----------+
|    101 |         3 |
|    102 |         1 |
|    103 |         1 |
+--------+-----------+
3 rows in set (0.04 sec)
```

TRANSACTIONS

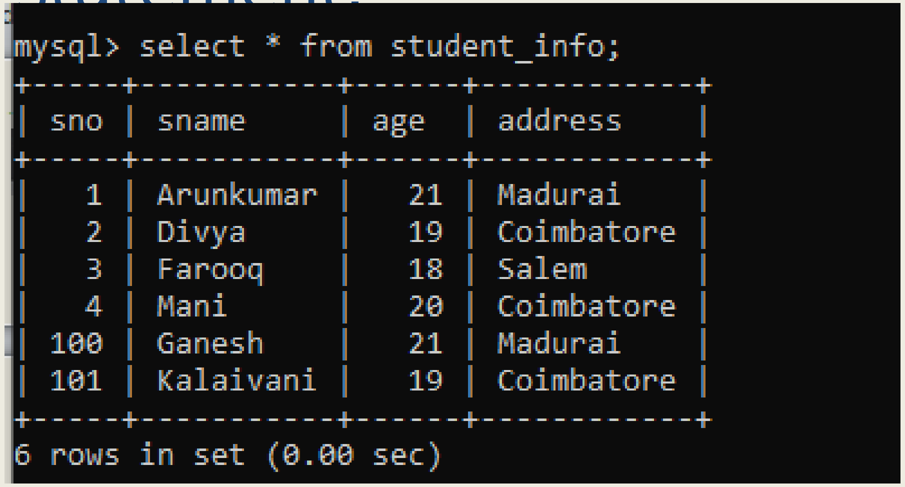
Transactions are **units or sequences of work accomplished in a logical order**, whether in a manual fashion by a user or automatically by some sort of a database program.

A database transaction is the propagation of one or more changes as a single action on the database.

- The **COMMIT statement** saves all the modifications made in the current.
- The **ROLLBACK operation** undoes all the changes done by the current transaction i.e. If you invoke this statement, all the modifications are reverted until the last RT TRANSACTION

TRANSACTIONS

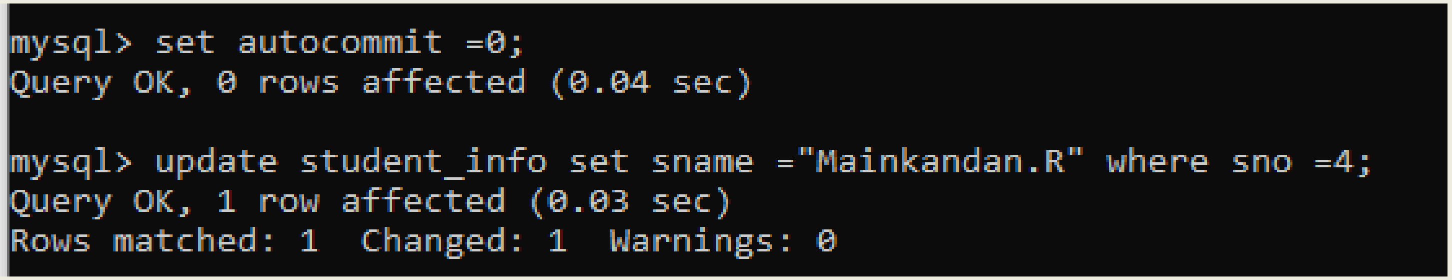
- The **ROLLBACK operation** undoes all the changes done by the current transaction i.e. If you invoke this statement, all the modifications are reverted until the last commit or the START TRANSACTION statement.



```
mysql> select * from student_info;
+-----+-----------+------+------------+
| sno | sname     | age  | address    |
+-----+-----------+------+------------+
|   1 | Arunkumar |   21 | Madurai    |
|   2 | Divya     |   19 | Coimbatore |
|   3 | Farooq    |   18 | Salem      |
|   4 | Mani      |   20 | Coimbatore |
| 100 | Ganesh    |   21 | Madurai    |
| 101 | Kalaivani |   19 | Coimbatore |
+-----+-----------+------+------------+
6 rows in set (0.00 sec)
```

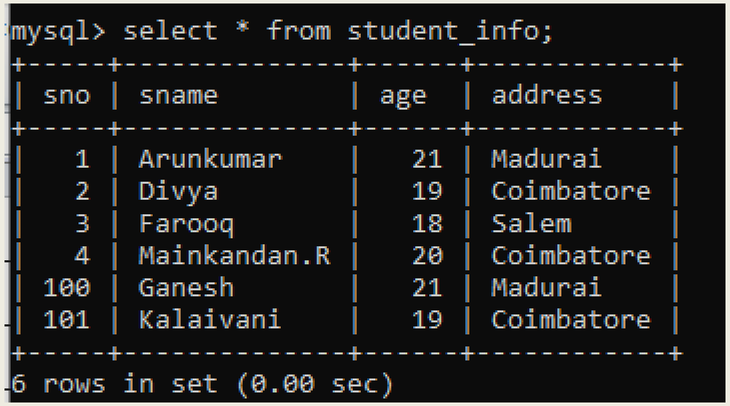
ROLLBACK operation

❖ **Update operation:**



```
mysql> set autocommit =0;
Query OK, 0 rows affected (0.04 sec)

mysql> update student_info set sname ="Mainkandan.R" where sno =4;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

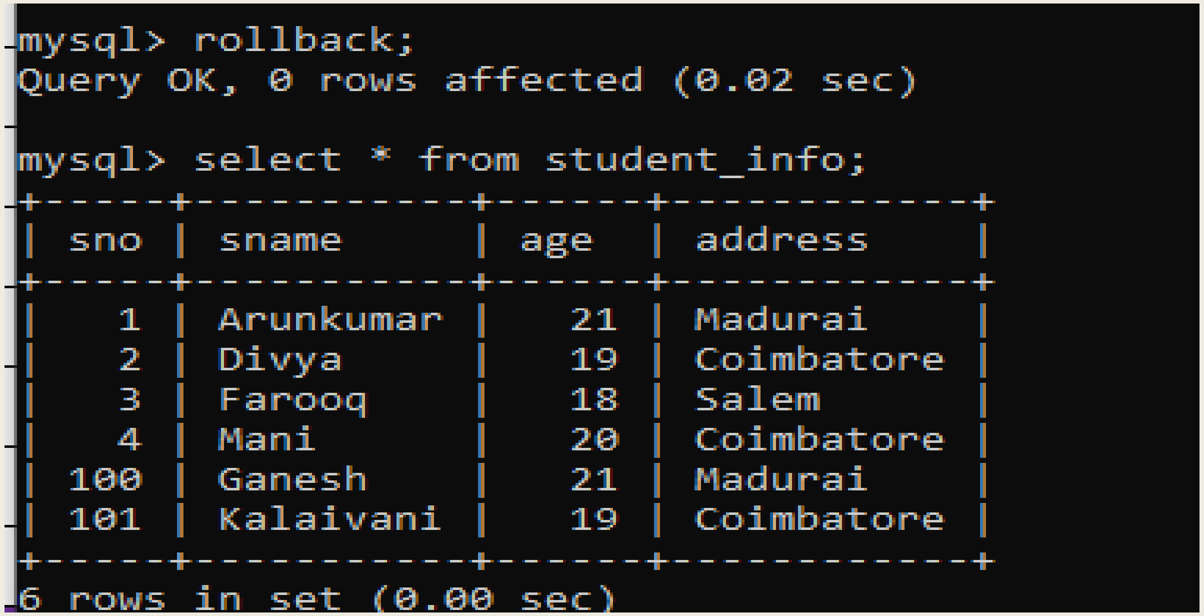
❖



```
mysql> select * from student_info;
+-----+--------------+------+------------+
| sno | sname        | age  | address    |
+-----+--------------+------+------------+
|   1 | Arunkumar    |   21 | Madurai    |
|   2 | Divya        |   19 | Coimbatore |
|   3 | Farooq       |   18 | Salem      |
|   4 | Mainkandan.R |   20 | Coimbatore |
| 100 | Ganesh       |   21 | Madurai    |
| 101 | Kalaivani    |   19 | Coimbatore |
+-----+--------------+------+------------+
6 rows in set (0.00 sec)
```

ROLLBACK operation

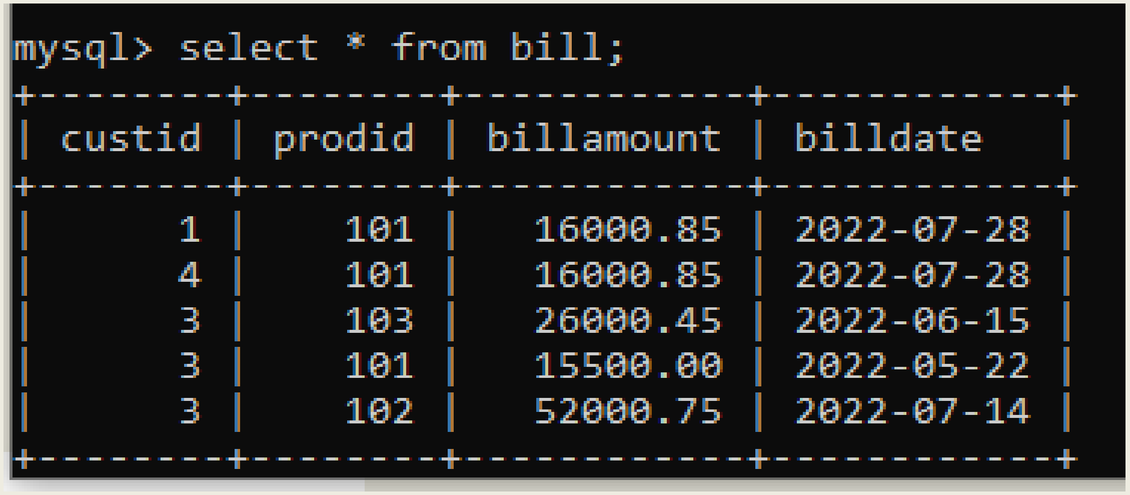
❖ **Rollback operation:**



```
mysql> rollback;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from student_info;
+------+-----------+-------+------------+
| sno  | sname     | age   | address    |
+------+-----------+-------+------------+
|    1 | Arunkumar |    21 | Madurai    |
|    2 | Divya     |    19 | Coimbatore |
|    3 | Farooq    |    18 | Salem      |
|    4 | Mani      |    20 | Coimbatore |
|  100 | Ganesh    |    21 | Madurai    |
|  101 | Kalaivani |    19 | Coimbatore |
+------+-----------+-------+------------+
6 rows in set (0.00 sec)
```

ROLLBACK operation

- After Rollback operation:

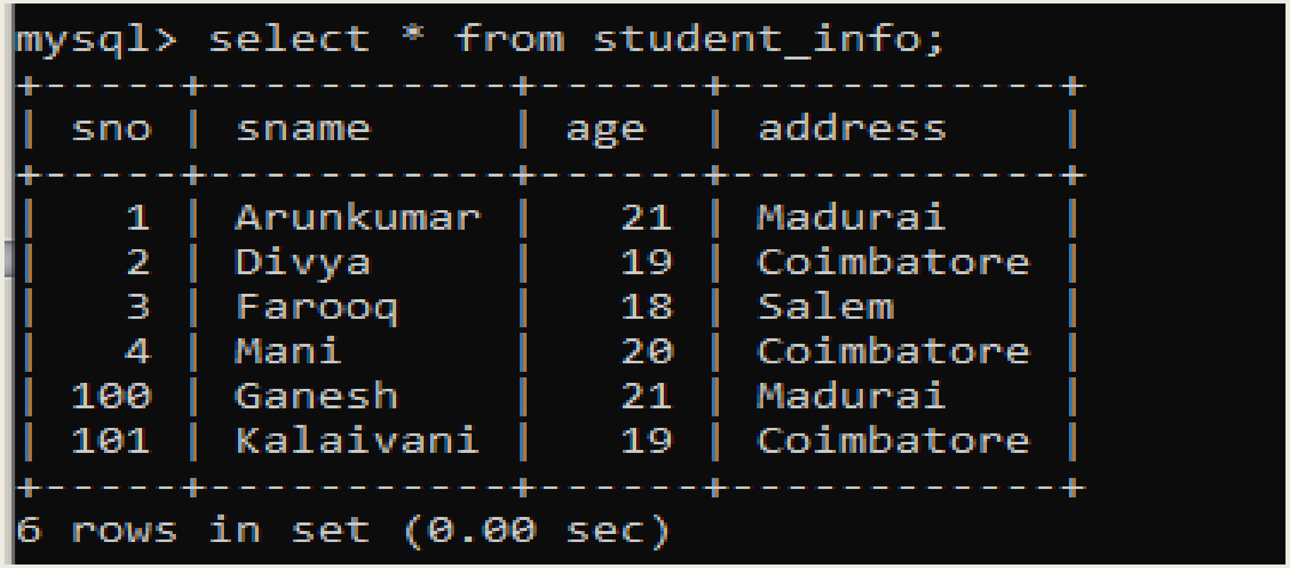


```
mysql> select * from bill;
+--------+--------+------------+------------+
| custid | prodid | billamount | billdate   |
+--------+--------+------------+------------+
|      1 |    101 |   16000.85 | 2022-07-28 |
|      4 |    101 |   16000.85 | 2022-07-28 |
|      3 |    103 |   26000.45 | 2022-06-15 |
|      3 |    101 |   15500.00 | 2022-05-22 |
|      3 |    102 |   52000.75 | 2022-07-14 |
+--------+--------+------------+------------+
```

- The update cannot be reflect on the database after rollback – (it restore the previous data)

- The **COMMIT statement** saves all the modifications made in the current.

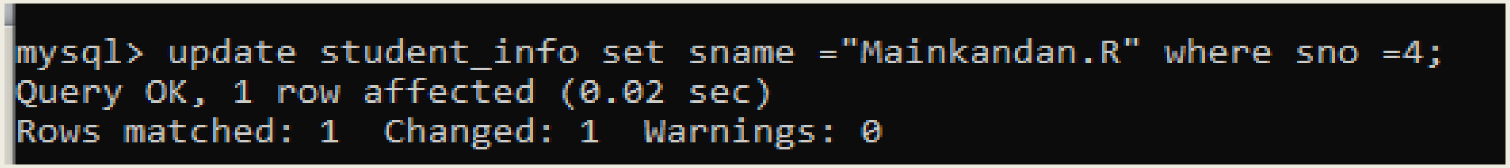
- BEFORE COMMIT:



```
mysql> select * from student_info;
+-----+-----------+------+------------+
| sno | sname     | age  | address    |
+-----+-----------+------+------------+
|   1 | Arunkumar |   21 | Madurai    |
|   2 | Divya     |   19 | Coimbatore |
|   3 | Farooq    |   18 | Salem      |
|   4 | Mani      |   20 | Coimbatore |
| 100 | Ganesh    |   21 | Madurai    |
| 101 | Kalaivani |   19 | Coimbatore |
+-----+-----------+------+------------+
6 rows in set (0.00 sec)
```

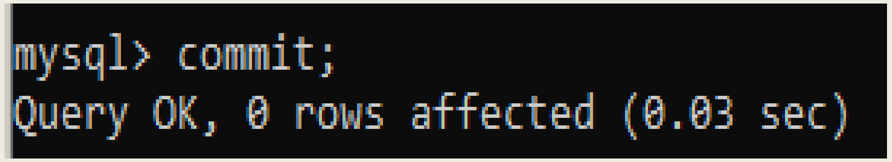
COMMIT operation

❖ **Update operation:**



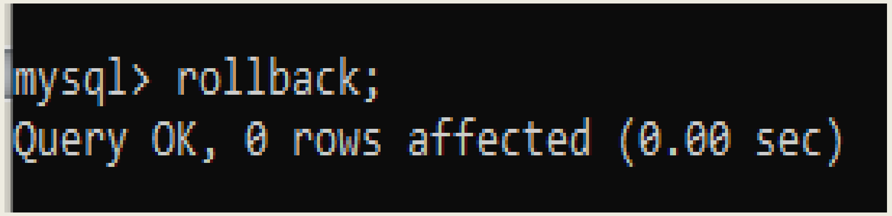
```
mysql> update student_info set sname ="Mainkandan.R" where sno =4;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

❖ **COMMIT OPERATION:**



```
mysql> commit;
Query OK, 0 rows affected (0.03 sec)
```

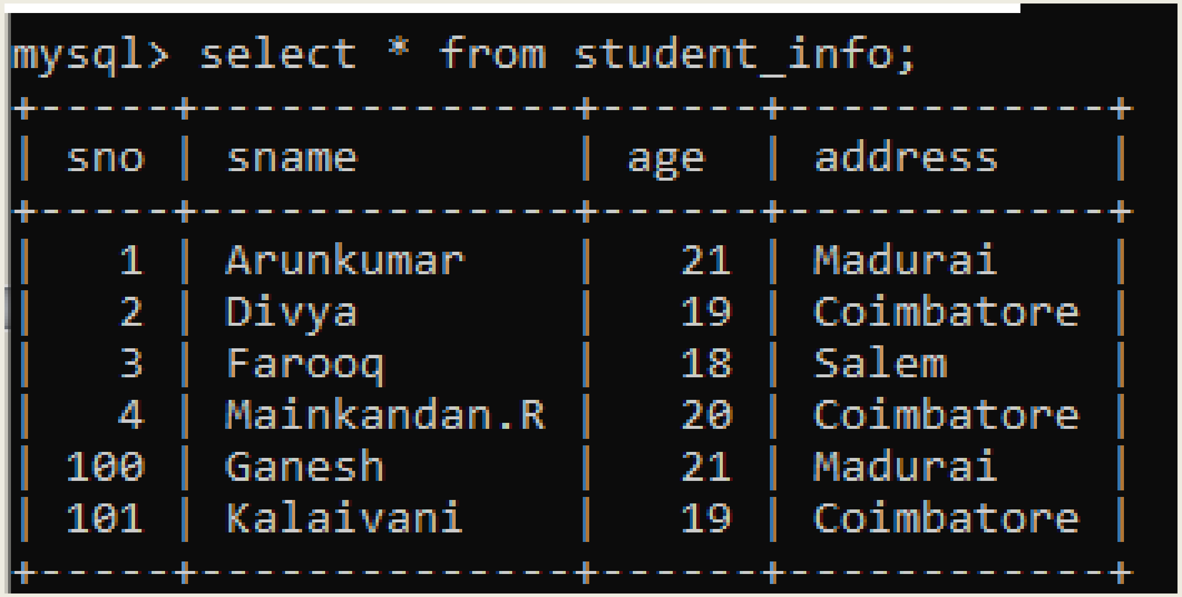
❖ **Rollback operation:**



```
mysql> rollback;
Query OK, 0 rows affected (0.00 sec)
```

COMMIT operation

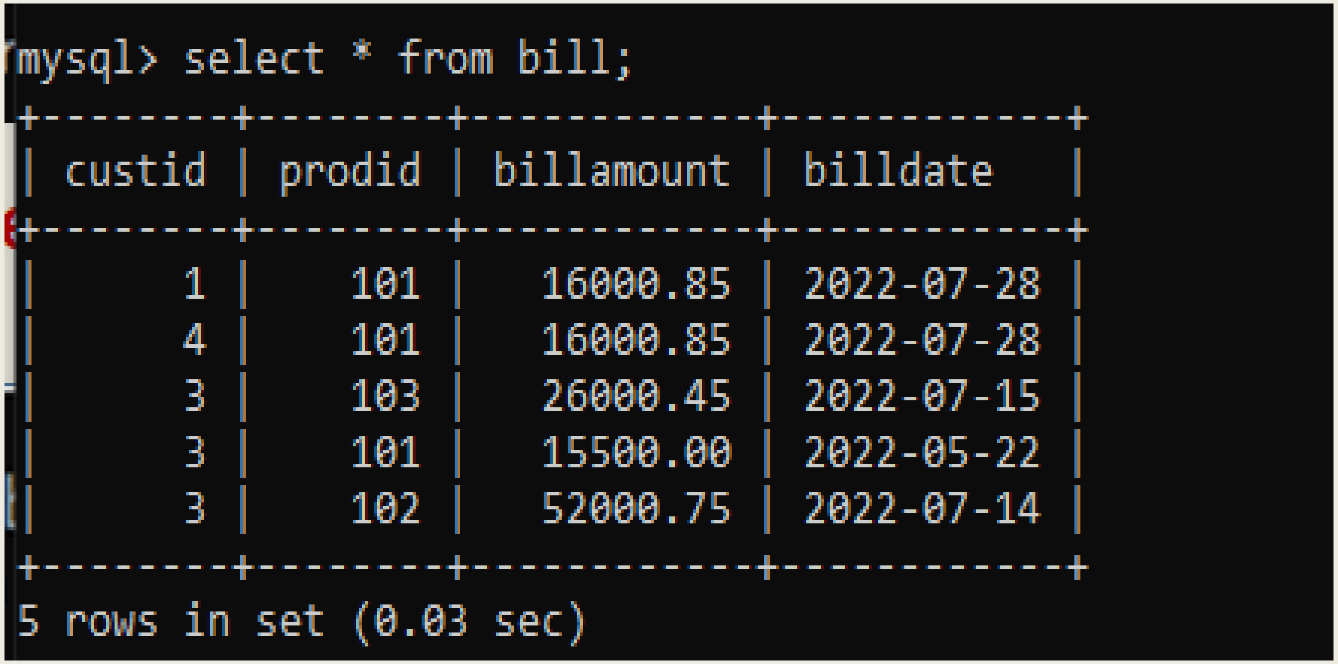
❖ **After Commit and Rollback operation:**



```
mysql> select * from student_info;
+-----+---------------+------+------------+
| sno | sname         | age  | address    |
+-----+---------------+------+------------+
|   1 | Arunkumar     |   21 | Madurai    |
|   2 | Divya         |   19 | Coimbatore |
|   3 | Farooq        |   18 | Salem      |
|   4 | Mainkandan.R  |   20 | Coimbatore |
| 100 | Ganesh        |   21 | Madurai    |
| 101 | Kalaivani     |   19 | Coimbatore |
+-----+---------------+------+------------+
```

COMMIT operation

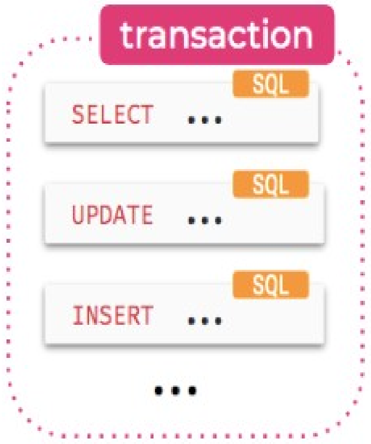
❖ **After COMMIT & Rollback operation :**



```
mysql> select * from bill;
+--------+--------+------------+------------+
| custid | prodid | billamount | billdate   |
+--------+--------+------------+------------+
|      1 |    101 |   16000.85 | 2022-07-28 |
|      4 |    101 |   16000.85 | 2022-07-28 |
|      3 |    103 |   26000.45 | 2022-07-15 |
|      3 |    101 |   15500.00 | 2022-05-22 |
|      3 |    102 |   52000.75 | 2022-07-14 |
+--------+--------+------------+------------+
5 rows in set (0.03 sec)
```

❖ **The data cannot be changed once committed and never rollback the previous data**

Transactions



A transaction is a logical group of one or more SQL statements.

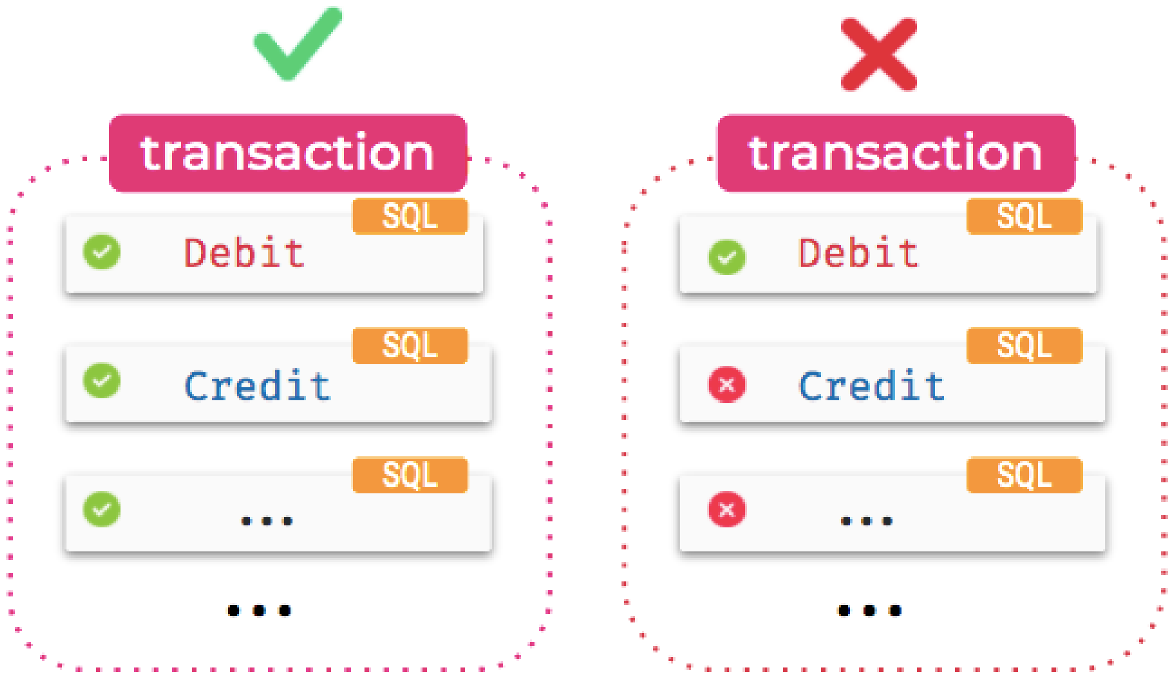
Transactions are used in various scenarios such as banking, ecommerce, social networks, booking tickets, etc.

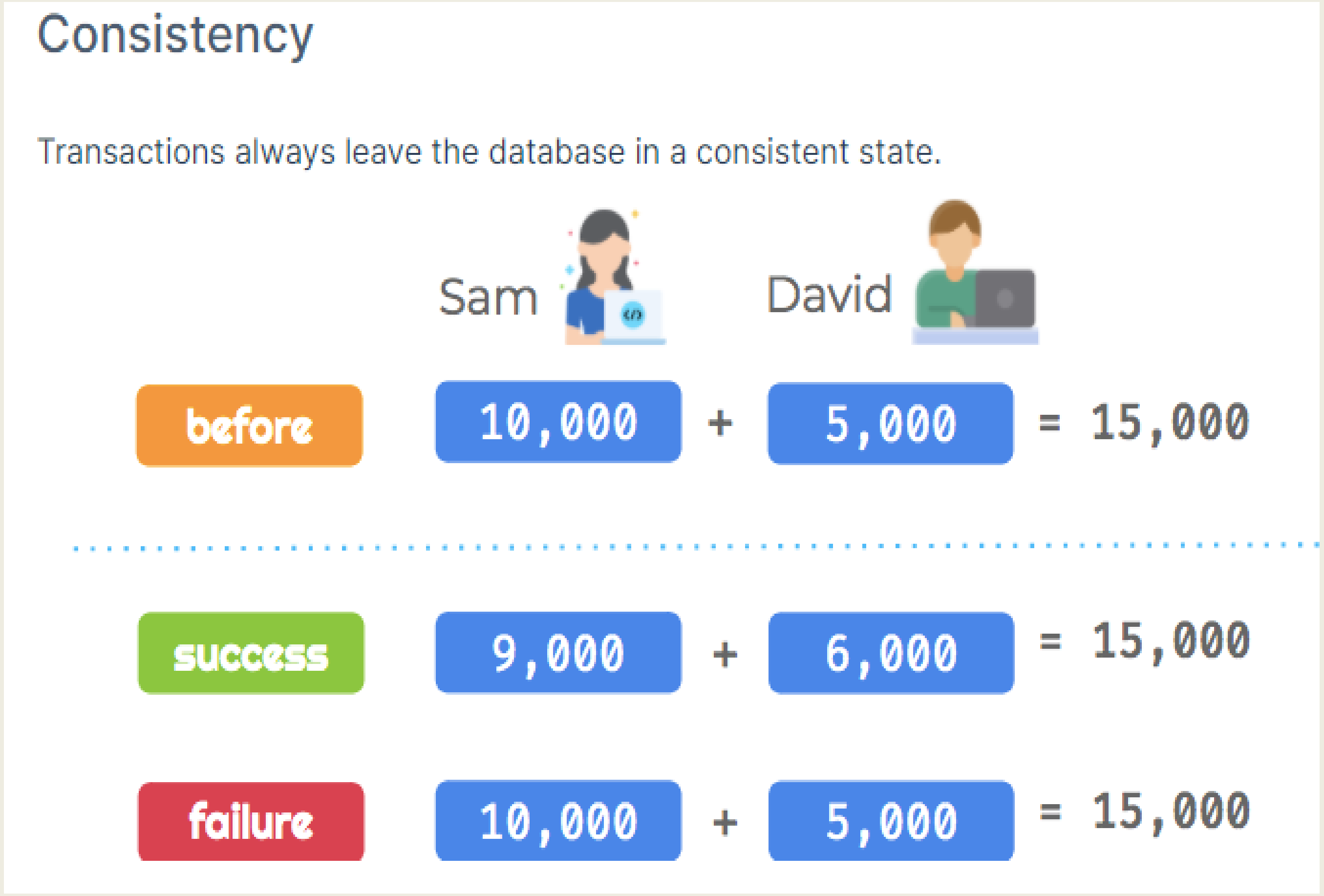
A transaction has four important properties.

- Atomicity
- Consistency
- Isolation
- Durability

Atomicity

Either all SQL statements or none are applied to the database.

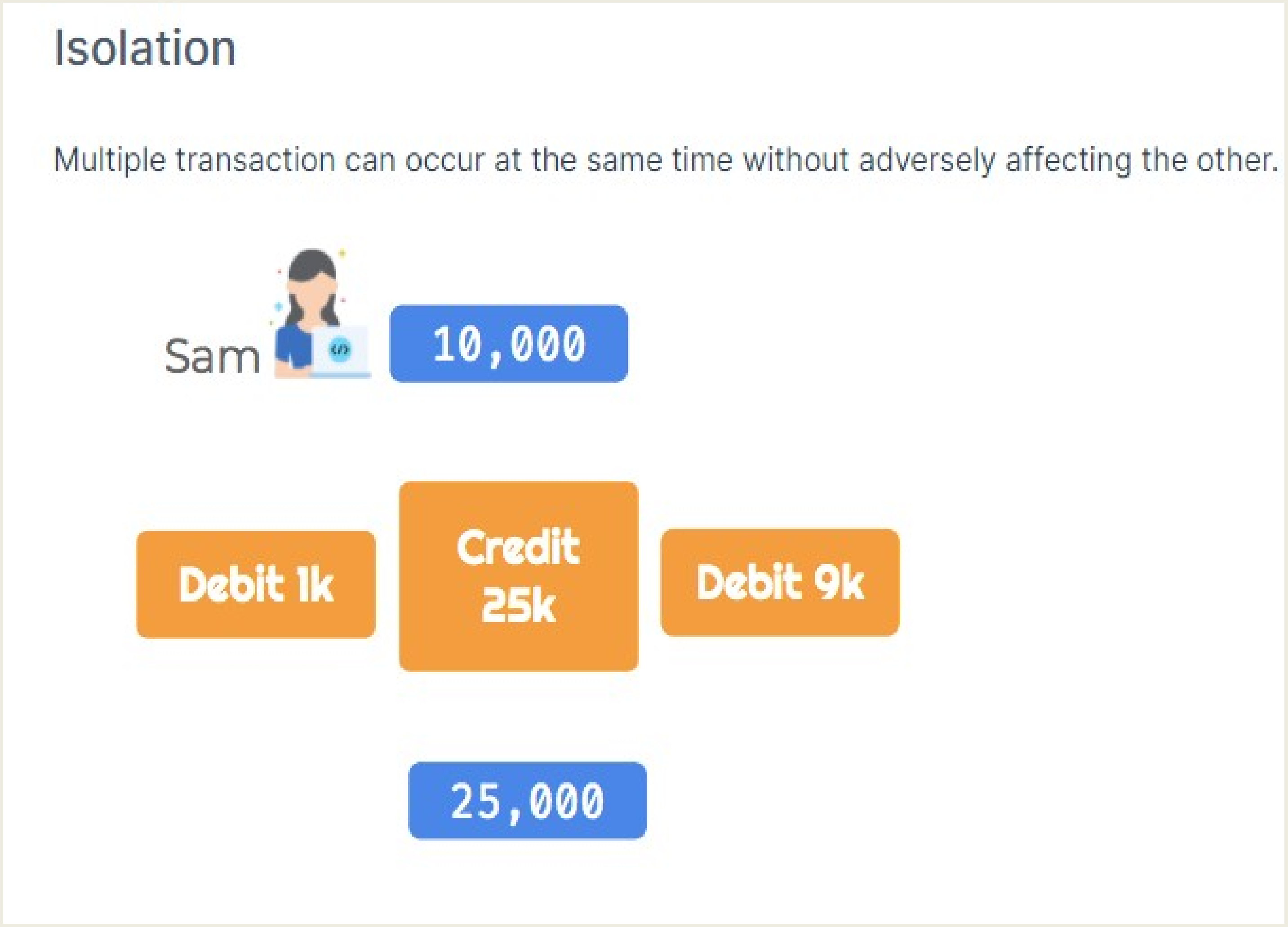




Consistency

Transactions always leave the database in a consistent state.

	Sam		David	
before	10,000	+	5,000	= 15,000
success	9,000	+	6,000	= 15,000
failure	10,000	+	5,000	= 15,000

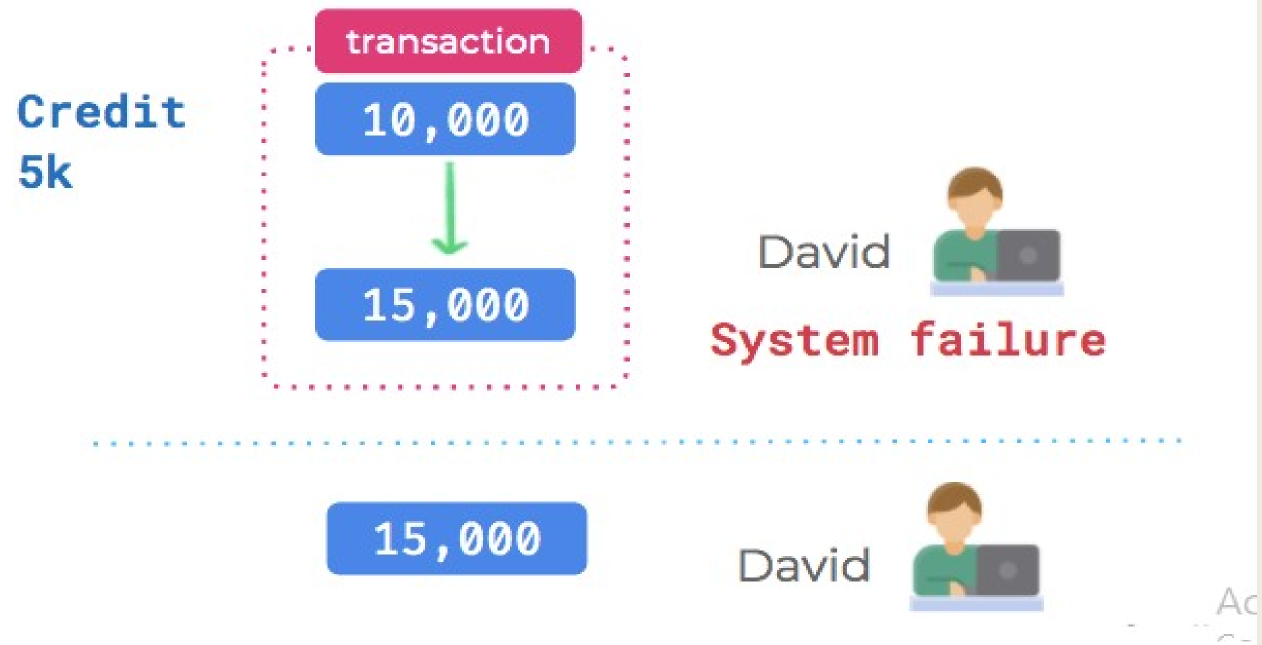


Isolation

Multiple transaction can occur at the same time without adversely affecting the other.

Durability

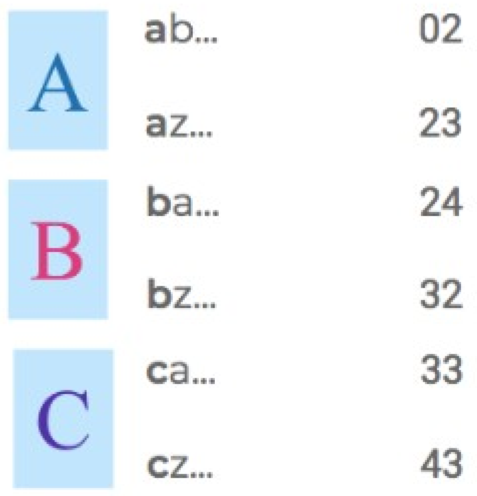
Changes of a successful transaction persist even after a system crash.



These four properties are commonly acronymed as ACID.

Atomicity **C**onsistency **I**solation **D**urable

Indexes



In scenarios like, searching for a word in dictionary, we use index to easily search for the word. Similarly, in databases, we maintain indexes to speed up the search for data in a table.

KGiSL MicroCollege

VIEWS

- A view is a database object that has no values. Its contents are based on the base table. It contains rows and columns similar to the real table.
- In MySQL, the View is a **virtual table** created by a query by joining one or more tables.
- It is operated similarly to the base table but does not contain any data of its own.
- The View and table have one main difference that the views are definitions built on top of other tables (or views).
- If any changes occur in the underlying table, the same changes reflected in the View also.

VIEWS

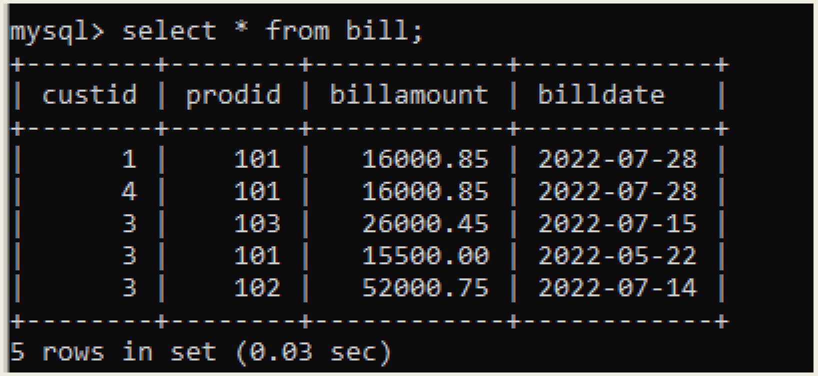
- ❑ A view is a database object that has no values. Its contents are based on the base table. It contains rows and columns similar to the real table.
- ❑ In MySQL, the View is a **virtual table** created by a query by joining one or more tables.
- ❑ It is operated similarly to the base table but does not contain any data of its own.
- ❑ The View and table have one main difference that the views are definitions built on top of other tables (or views).
- ❑ If any changes occur in the underlying table, the same changes reflected in the View also.

You can use views **to hide table columns from users by granting them access to the view and not to the table itself**.

KGiSL MicrCollege

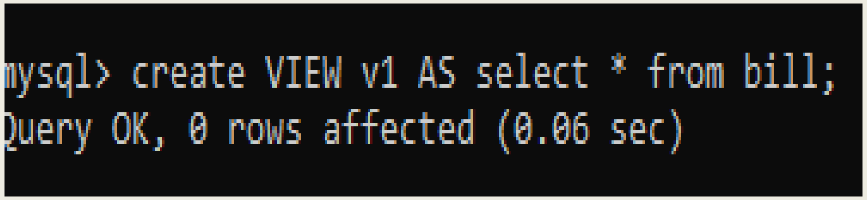
CREATE VIEW:

TABLE:

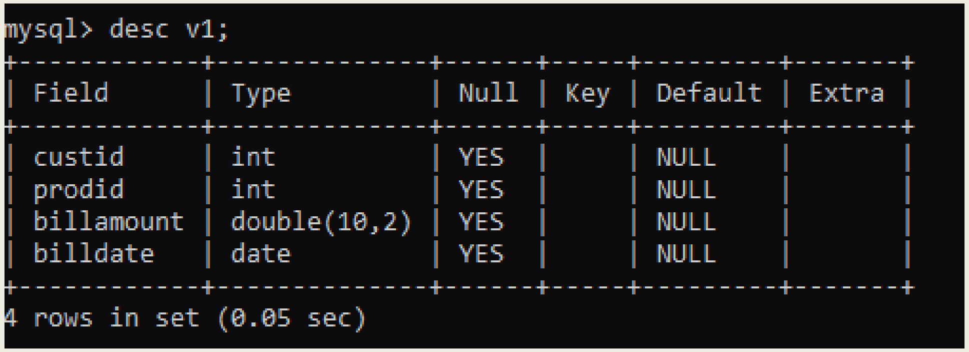


```
mysql> select * from bill;
+--------+--------+------------+------------+
| custid | prodid | billamount | billdate   |
+--------+--------+------------+------------+
|      1 |    101 |   16000.85 | 2022-07-28 |
|      4 |    101 |   16000.85 | 2022-07-28 |
|      3 |    103 |   26000.45 | 2022-07-15 |
|      3 |    101 |   15500.00 | 2022-05-22 |
|      3 |    102 |   52000.75 | 2022-07-14 |
+--------+--------+------------+------------+
5 rows in set (0.03 sec)
```

VIEW CREATION:

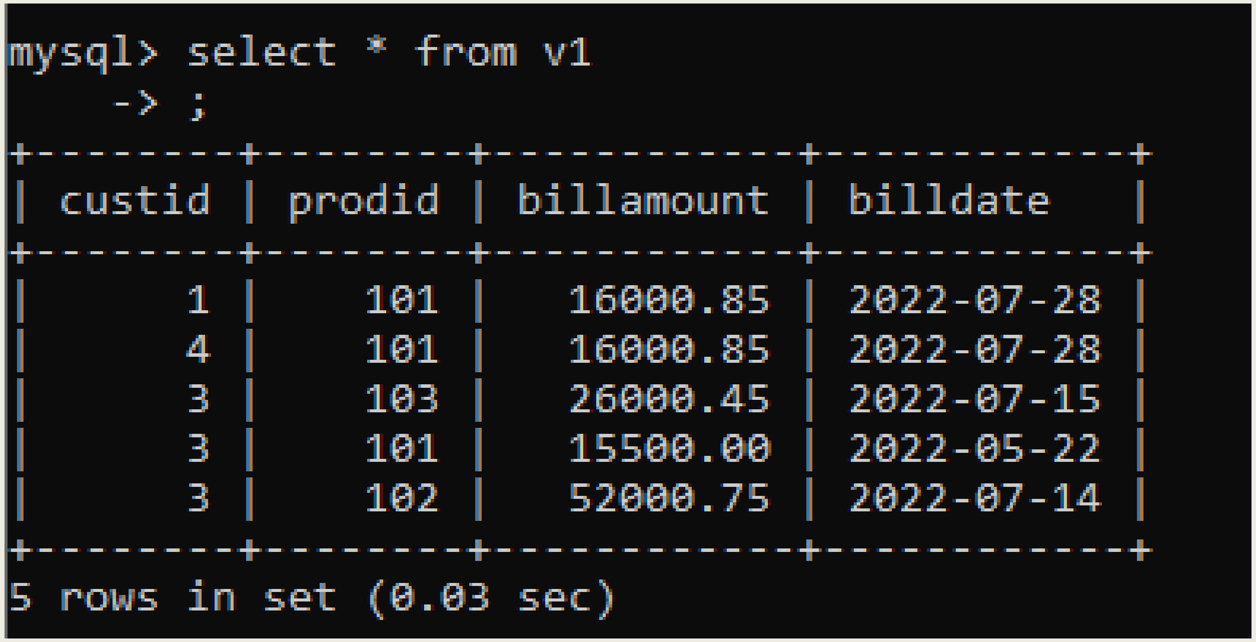


```
mysql> create VIEW v1 AS select * from bill;
Query OK, 0 rows affected (0.06 sec)
```



```
mysql> desc v1;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| custid     | int          | YES  |     | NULL    |       |
| prodid     | int          | YES  |     | NULL    |       |
| billamount | double(10,2) | YES  |     | NULL    |       |
| billdate   | date         | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
4 rows in set (0.05 sec)
```

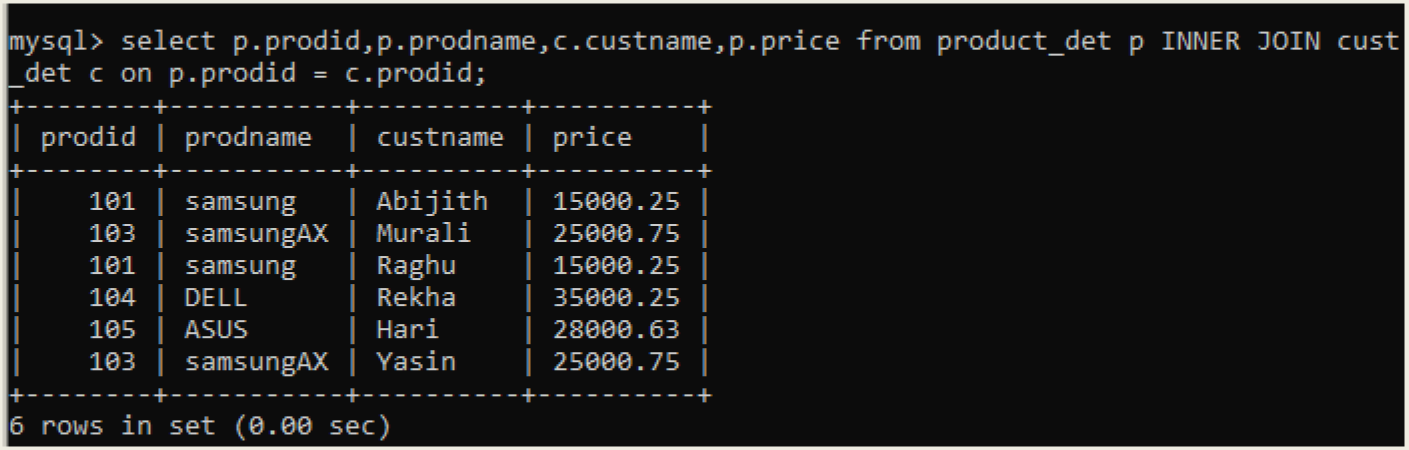
SELECT QUERY USING VIEW:



```
mysql> select * from v1
    -> ;
+--------+--------+------------+------------+
| custid | prodid | billamount | billdate   |
+--------+--------+------------+------------+
|      1 |    101 |   16000.85 | 2022-07-28 |
|      4 |    101 |   16000.85 | 2022-07-28 |
|      3 |    103 |   26000.45 | 2022-07-15 |
|      3 |    101 |   15500.00 | 2022-05-22 |
|      3 |    102 |   52000.75 | 2022-07-14 |
+--------+--------+------------+------------+
5 rows in set (0.03 sec)
```

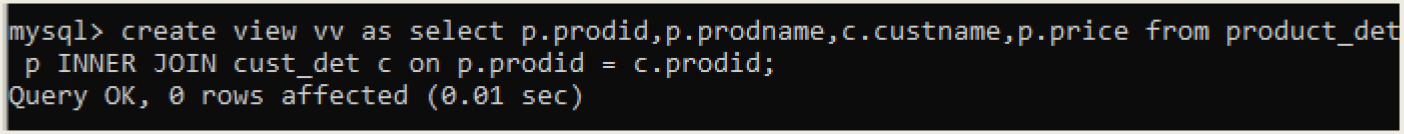
CREATE VIEW FOR JOINS:

JOIN TABLE:



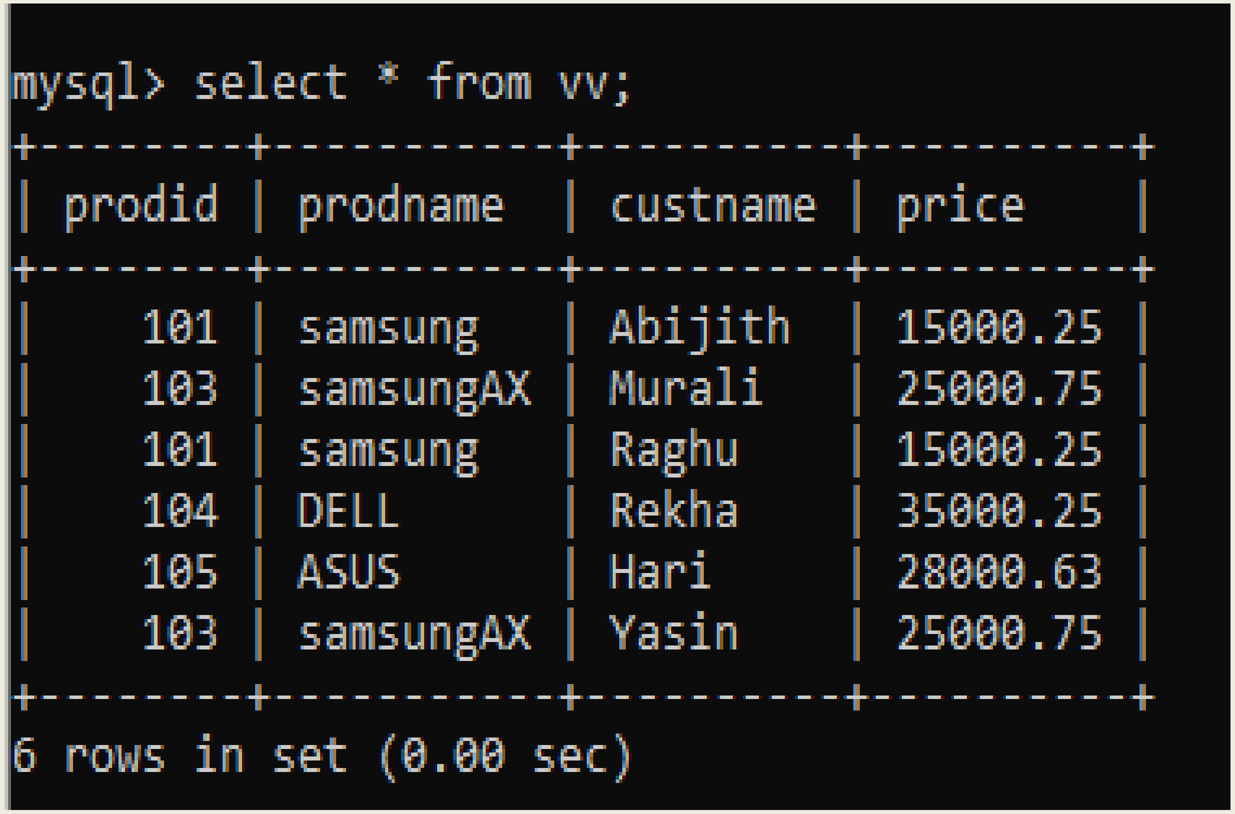
```
mysql> select p.prodid,p.prodname,c.custname,p.price from product_det p INNER JOIN cust
_det c on p.prodid = c.prodid;
+--------+-----------+----------+----------+
| prodid | prodname  | custname | price    |
+--------+-----------+----------+----------+
|    101 | samsung   | Abijith  | 15000.25 |
|    103 | samsungAX | Murali   | 25000.75 |
|    101 | samsung   | Raghu    | 15000.25 |
|    104 | DELL      | Rekha    | 35000.25 |
|    105 | ASUS      | Hari     | 28000.63 |
|    103 | samsungAX | Yasin    | 25000.75 |
+--------+-----------+----------+----------+
6 rows in set (0.00 sec)
```

VIEW CREATION:



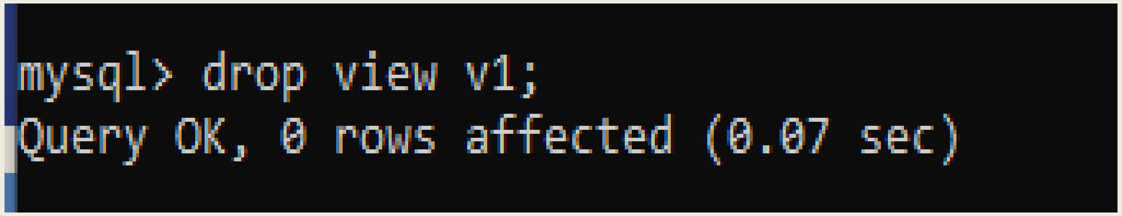
```
mysql> create view vv as select p.prodid,p.prodname,c.custname,p.price from product_det
 p INNER JOIN cust_det c on p.prodid = c.prodid;
Query OK, 0 rows affected (0.01 sec)
```

SELECT JOIN QUERY USING VIEW:

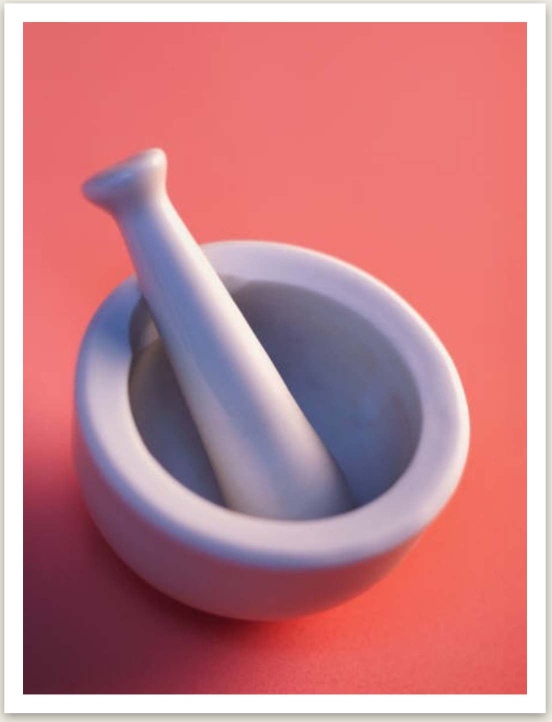


```
mysql> select * from vv;
+--------+-----------+----------+----------+
| prodid | prodname  | custname | price    |
+--------+-----------+----------+----------+
|    101 | samsung   | Abijith  | 15000.25 |
|    103 | samsungAX | Murali   | 25000.75 |
|    101 | samsung   | Raghu    | 15000.25 |
|    104 | DELL      | Rekha    | 35000.25 |
|    105 | ASUS      | Hari     | 28000.63 |
|    103 | samsungAX | Yasin    | 25000.75 |
+--------+-----------+----------+----------+
6 rows in set (0.00 sec)
```

DROP VIEW:



```
mysql> drop view v1;
Query OK, 0 rows affected (0.07 sec)
```



Thank You