# University of Central Florida
# CGS 2545
# Database Concepts

DEPARTMENT OF ELECTRICAL ENGINEERING & COMPUTER SCIENCE

COMPUTER SCIENCE DIVISION

# Transactions

- A transaction is a unit of work that is performed against a database.
- 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 transaction is the propagation of one or more changes to the database.

# Transactions

- For example, if you are creating a record or updating a record or deleting a record from the table, then you are performing a transaction on that table.
- It is important to control these transactions to ensure the data integrity and to handle database errors.
- Practically, you will club many SQL queries into a group and you will execute all of them together as a part of a transaction.

# Transactions

- Properties of Transactions
  - Transactions have the following four standard properties, usually referred to by the acronym **ACID**.
    - **Atomicity**
      - ensures that all operations within the work unit are completed successfully.
      - Otherwise, the transaction is aborted at the point of failure and all the previous operations are rolled back to their former state.

# Transactions

- Properties of Transactions
  - Transactions have the following four standard properties, usually referred to by the acronym **ACID**.
    - **Consistency**
      - ensures that the database properly changes states upon a successfully committed transaction.
    - **Isolation**
      - enables transactions to operate independently of and transparent to each other.
    - **Durability**
      - ensures that the result or effect of a committed transaction persists in case of a system failure.

# Transactions

- Transaction Control
  - The following commands are used to control transactions.
    - **COMMIT** – to save the changes.
    - **ROLLBACK** – to roll back the changes.
    - **SAVEPOINT** – creates points within the groups of transactions in which to ROLLBACK.
    - **SET TRANSACTION** – Places a name on a transaction.

# Transactions

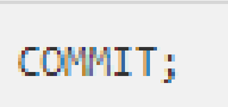
- Transactional Control Commands
  - Transactional control commands are only used with the **DML Commands** such as - INSERT, UPDATE and DELETE only.
  - They cannot be used while creating tables or dropping them because these operations are automatically committed in the database.

# Transactions

- Transactional Control Commands
  - The COMMIT Command
    - The COMMIT command is the transactional command used to save changes invoked by a transaction to the database.
    - The COMMIT command is the transactional command used to save changes invoked by a transaction to the database.
    - The COMMIT command saves all the transactions to the database since the last COMMIT or ROLLBACK command.

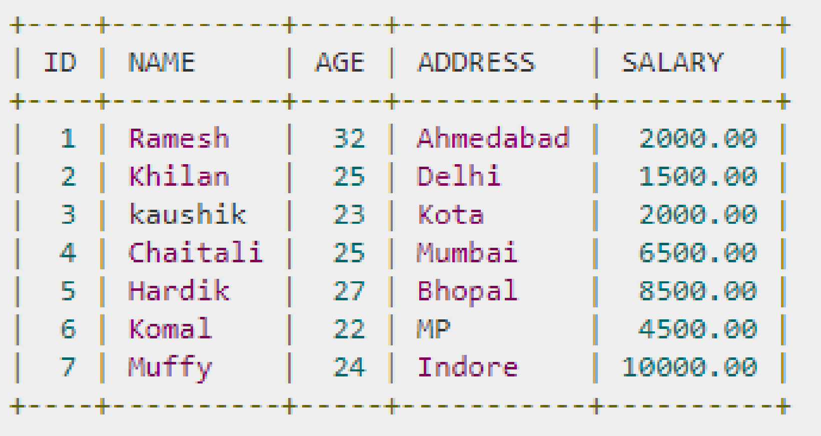
# Transactions

- Transactional Control Commands
  - The syntax for the COMMIT command is as follows.



```
COMMIT;
```

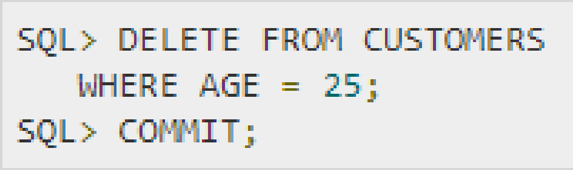
- Consider the CUSTOMERS table having the following records



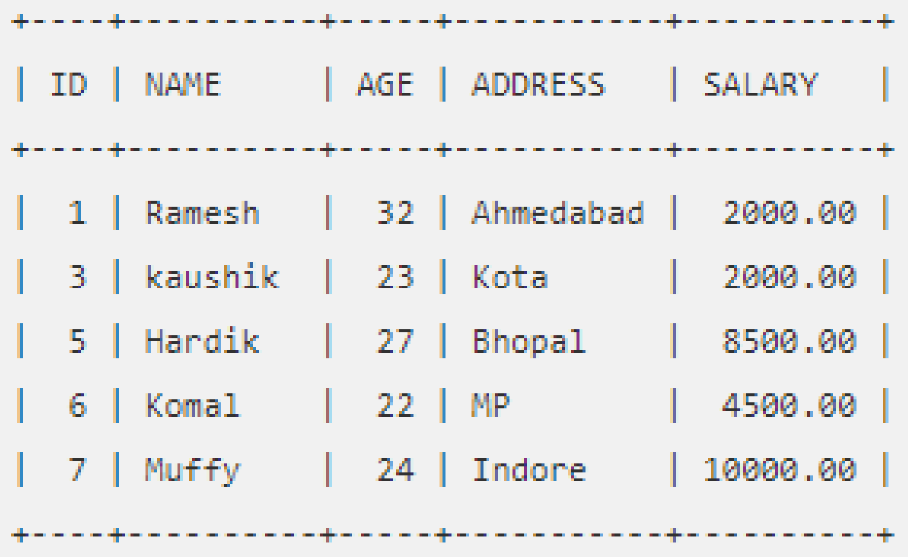
```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

# Transactions

- Transactional Control Commands
  - Following is an example which would delete those records from the table which have age = 25 and then COMMIT the changes in the database.



```
SQL> DELETE FROM CUSTOMERS
   WHERE AGE = 25;
SQL> COMMIT;
```



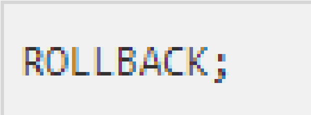
```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

# Transactions

- Transactional Control Commands
  - The ROLLBACK Command
    - The ROLLBACK command is the transactional command used to undo transactions that have not already been saved to the database.
    - This command can only be used to undo transactions since the last COMMIT or ROLLBACK command was issued.

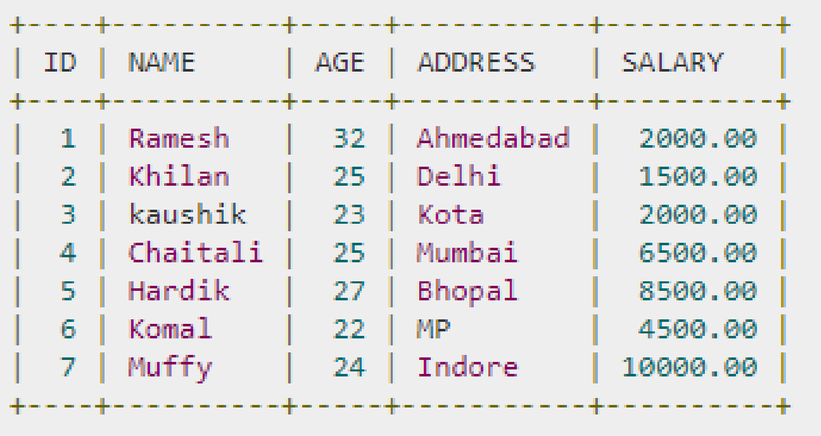
# Transactions

- Transactional Control Commands
  - The syntax for a ROLLBACK command is as follows



```
ROLLBACK;
```

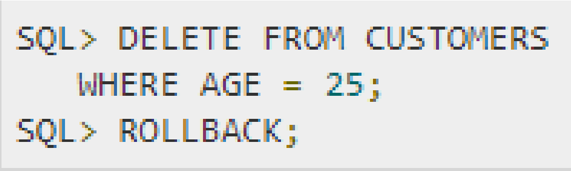
- Consider the CUSTOMERS table having the following records



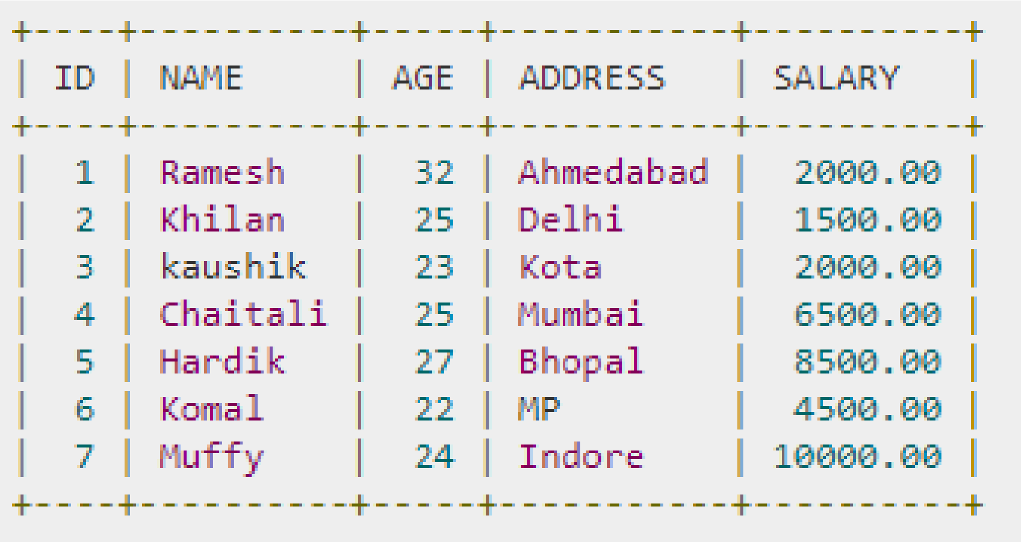
```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

# Transactions

- Transactional Control Commands
  - Following is an example which would delete those records from the table which have age = 25 and then ROLLBACK the changes in the database.



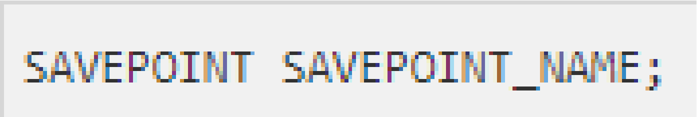
```
SQL> DELETE FROM CUSTOMERS
   WHERE AGE = 25;
SQL> ROLLBACK;
```



```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

# Transactions

- Transactional Control Commands
  - The SAVEPOINT Command
    - A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.
    - The syntax for a SAVEPOINT command is as shown below

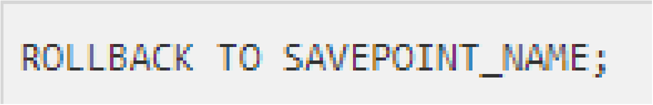


```
SAVEPOINT SAVEPOINT_NAME;
```

    - This command serves only in the creation of a SAVEPOINT among all the transactional statements.
    - The ROLLBACK command is used to undo a group of transactions.

# Transactions

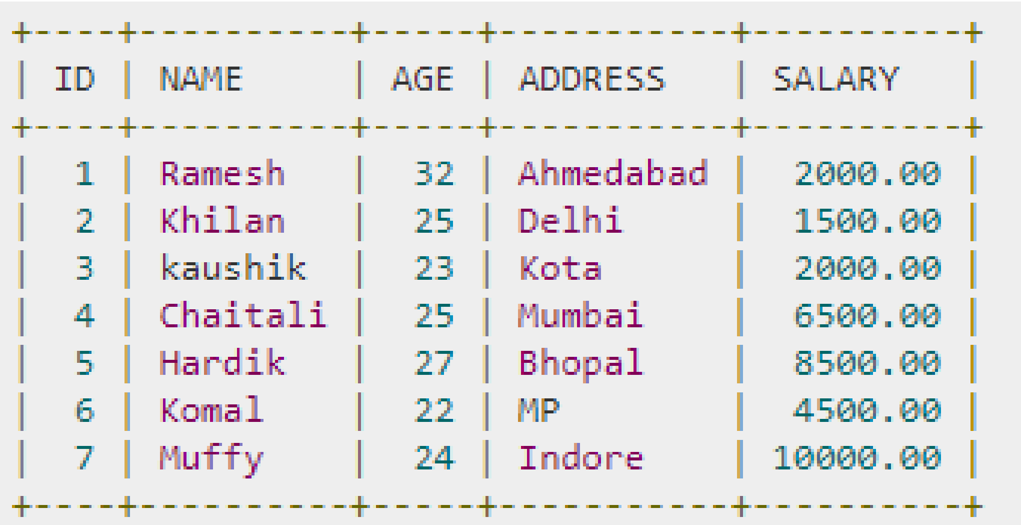
- Transactional Control Commands
  – The SAVEPOINT Command
    - The syntax for rolling back to a SAVEPOINT is as shown below.



```
ROLLBACK TO SAVEPOINT_NAME;
```

# Transactions

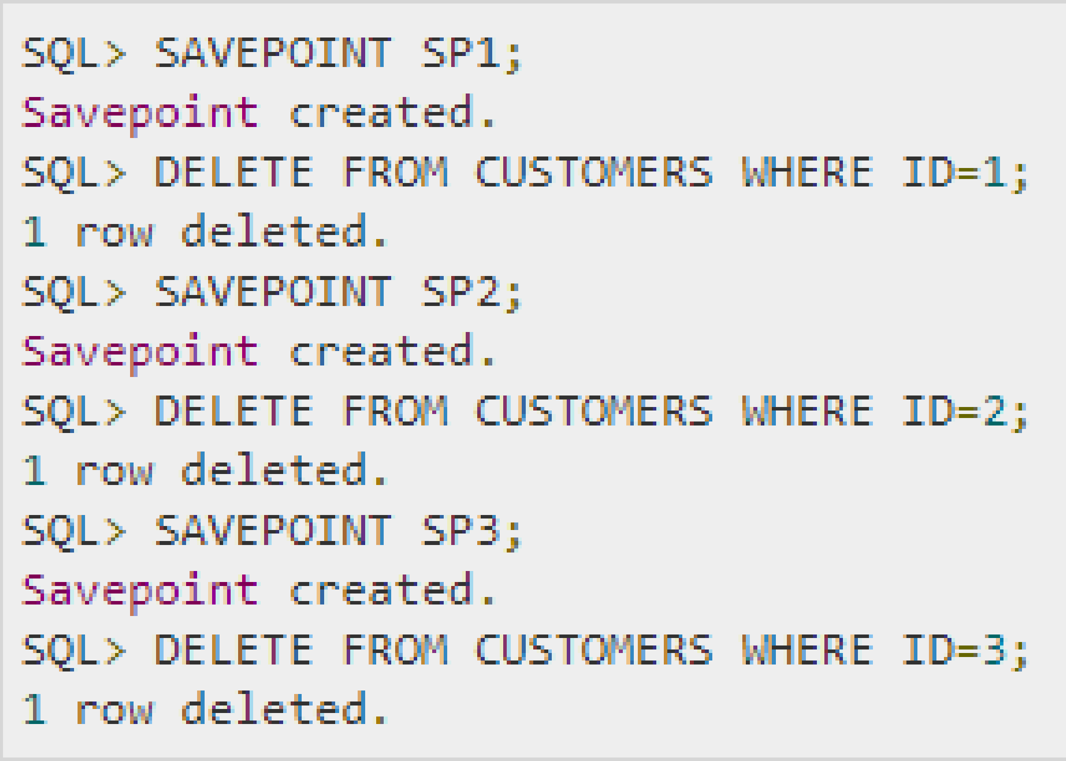
- Transactional Control Commands
  - The SAVEPOINT Command
    - Following is an example to delete three different records from the CUSTOMERS table.
    - Create a SAVEPOINT before each delete, that can be rolled back to at any time to return the appropriate data to its original state.



```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

# Transactions

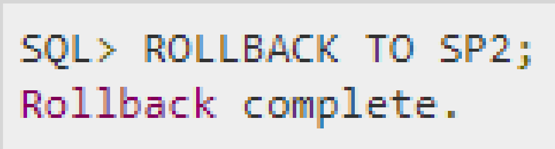
- Transactional Control Commands
  - The SAVEPOINT Command



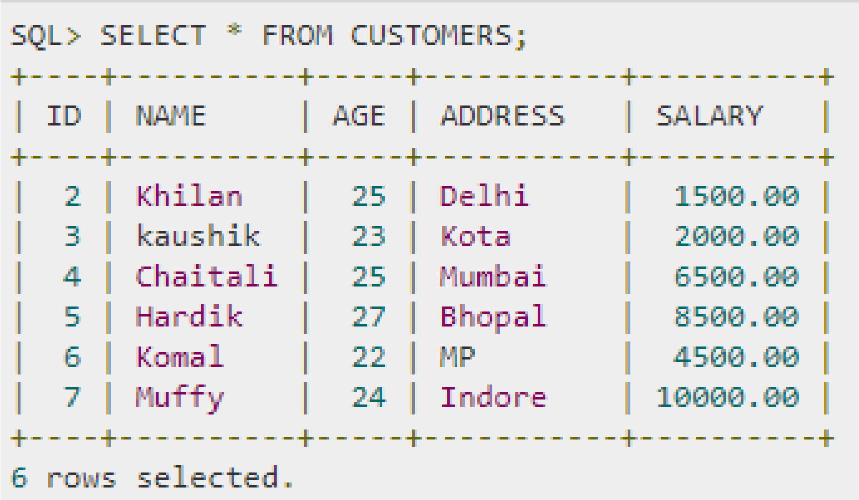
```
SQL> SAVEPOINT SP1;
Savepoint created.
SQL> DELETE FROM CUSTOMERS WHERE ID=1;
1 row deleted.
SQL> SAVEPOINT SP2;
Savepoint created.
SQL> DELETE FROM CUSTOMERS WHERE ID=2;
1 row deleted.
SQL> SAVEPOINT SP3;
Savepoint created.
SQL> DELETE FROM CUSTOMERS WHERE ID=3;
1 row deleted.
```

# Transactions

- Transactional Control Commands
  - The SAVEPOINT Command
    - Now that the three deletions have taken place, ROLLBACK to the SAVEPOINT that identified as SP2.
    - Because SP2 was created after the first deletion, the last two deletions are undone



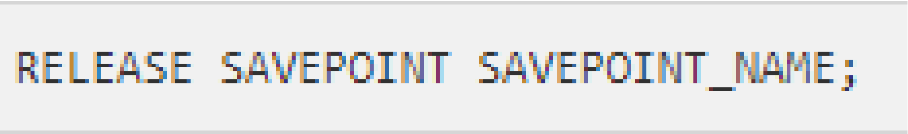
```
SQL> ROLLBACK TO SP2;
Rollback complete.
```



```
SQL> SELECT * FROM CUSTOMERS;
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
6 rows selected.
```

# Transactions

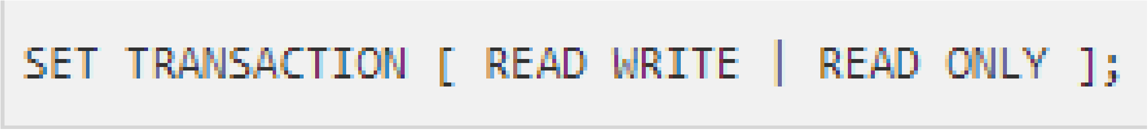
- Transactional Control Commands
  - The RELEASE SAVEPOINT Command
    - The RELEASE SAVEPOINT command is used to remove a SAVEPOINT that you have created.
    - Once a SAVEPOINT has been released, you can no longer use the ROLLBACK command to undo transactions performed since the last SAVEPOINT.
    - The syntax for a RELEASE SAVEPOINT command is as follows.



```
RELEASE SAVEPOINT SAVEPOINT_NAME;
```

# Transactions

- Transactional Control Commands
  - The SET TRANSACTION Command
    - The SET TRANSACTION command can be used to initiate a database transaction.
    - This command is used to specify characteristics for the transaction that follows.
    - For example, you can specify a transaction to be read only or read write.
    - The syntax for a SET TRANSACTION command is as follows.



```
SET TRANSACTION [ READ WRITE | READ ONLY ];
```