



TechM Full Stack Software Development

Course: Foundation of
Databases

Lecture On: Basic SQL
Queries

Instructor: Vishwa Mohan



Today's Agenda

1. **SQL queries:**
 - ALTER
 - DELETE
 - TRUNCATE
 - DROP
2. **Transactions:** *How they are supported in RDBMS*
3. **Transaction Control Language:** *SQL for transactions*
4. **Constraints:** *Why they are needed*
5. **Keys:** *Inside-out view*



The **ALTER** command is used at the following levels:

1. **Column level:** Used to add, delete and modify a column
2. **Table level:** Used to modify a table name
3. **Constraint Level:** Used to add or drop a constraint

If you need to alter a table frequently, then it means that you did not design it properly in the first place.



At the column level, the **ALTER** command can be used to:

1. Add a new column to a table

The syntax for this operation is as follows:

```
ALTER TABLE table_name  
ADD column_name datatype;
```

2. Modify an existing column

The syntax for this operation is as follows:

```
ALTER TABLE table_name  
MODIFY column_name datatype;
```

Renaming a column:

```
ALTER TABLE table_name  
RENAME column_name datatype TO new_name;
```



At the Column level, the ALTER command can be used to:

3. Delete a column from a table

The syntax for this operation is as follows:

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

Let's take a look at Example 1.



At the table level, the ALTER command can be used to:

1. Modify a table name

The syntax for this operation is as follows:

```
ALTER TABLE table_name  
RENAME TO new_table_name ;
```

Let's take a look at Example 2.

ALTER Command Hands-On Exercise (10 min)

Let's perform the following operations on the 'movie_theatre' table created previously:

1. Add two new columns named 'ticket_price' of type NUMBER(10) and 'location' of type VARCHAR2(12).
2. Update the type of the 'ticket_price' column to NUMBER(5,2) from NUMBER(10) for this newly added column.
3. Delete the column location from this table movie_theatre.
4. Rename the table from movie_theatre to theatre.



The **DELETE** command is used to delete the existing records from a table.

The syntax DELETE commands is as follows:

```
DELETE FROM table_name  
WHERE condition;
```

*If you do not provide the **where** condition, then it will delete all the records in the table.*

Let's take a look at Example 3.

DELETE Command Hands-On Exercise (5 min)

Let's perform the following operations on the 'movie' table:

1. Delete the record of movie named *Life of Pie*
2. Delete all records

Coding Example (5 min)

A table named 'inventory' stores a list of the items in a fruit store's inventory. The table has the following structure:

item_number	CHAR(5)
item_name	VARCHAR2(10)
item_desc	VARCHAR2(50)
item_count	NUMBER(5)

Three records are stored in the table. The store operations underwent many modifications, and now, the structure of the table needs to undergo the following modifications:

1. Rename the item_count column as inventory_count
2. Add a column named 'last_updated' of type DATE.
3. Delete all existing records

Let's take a look at Coding Example (Inventory).

Poll 1

Which of the following is a DDL statement?

1. ALTER
2. INSERT
3. DELETE
4. None of the above

Poll 1 (Answer)

Which of the following is a DDL statement?

1. ALTER

2. INSERT

3. DELETE

4. None of the above



The TRUNCATE command is used for the following purposes:

For completely deleting the data in a table but not the table itself.

The syntax for this operation is as follows:

```
TRUNCATE TABLE table_name;
```

TRUNCATE works similarly to the delete query without a where clause. However, TRUNCATE cannot be used with the where clause.

Let's take a look at Example 4.

TRUNCATE Command Hands-On Exercise (2 min)

Truncate 'customer' table.





Truncation

Vs



TRUNCATE

1. Data Definition Language
2. It recovers space occupied by the records in the database
3. It can be used to delete all the records in a table

DELETE

1. Data Manipulation Language
2. It does not recover space occupied by the records in the database even after the records are deleted
3. It can be used to delete the records partially from a table



The DROP command is used to delete a table in a database.

The syntax for this operation is as follows:

```
DROP TABLE table_name;
```

Be careful! It leads to loss of the table altogether from the database.

Let's take a look at Example 5.

**TRUNCATE**

1. Data Definition Language
2. It cannot be rolled back.
3. It recovers space.
4. It can be used to delete all the records in a table

**DELETE**

1. Data Manipulation Language
2. It can be rolled back.
3. It does not recover space.
4. It can be used to delete the records partially in a table.

**DROP**

1. Data Definition Language
2. It cannot be rolled back.
3. It completely removes a table from the database

Poll 2

Which of the following commands can be used to delete all the records in a table?

1. TRUNCATE
2. ALTER
3. DELETE
4. Both 1 and 3

Poll 2 (Answer)

Which of the following commands can be used to delete all the records in a table?

1. TRUNCATE

2. ALTER

3. DELETE

4. Both 1 and 3

Poll 3

Which of the following commands can be used to delete all the records in a table without deleting the table itself?

1. TRUNCATE
2. DROP
3. DELETE
4. Both 1 and 3

Poll 3 (Answer)

Which of the following commands can be used to delete all the records in a table without deleting the table itself?

1. TRUNCATE

2. DROP

3. DELETE

4. Both 1 and 3

Poll 4

Which of the following commands can be used to delete all the records in a table as well as the table itself?

1. TRUNCATE
2. DROP
3. DELETE
4. None of the above

Poll 4 (Answer)

Which of the following commands can be used to delete all the records in a table as well as the table itself?

1. TRUNCATE

2. DROP

3. DELETE

4. None of the above

01

A database built using a relational model is called a relational database. A relational database management system manages relational databases.

02

A transaction in a relational database is one whole process that cannot be divided into smaller processes.

03

If a user transfers money from one account to another, then this activity either does not happen at all or is completed.

04

If money is deducted from one account, then it must be received by the other.

Atomicity

Every transaction is atomic. It is either completed or does not happen at all.

Consistency

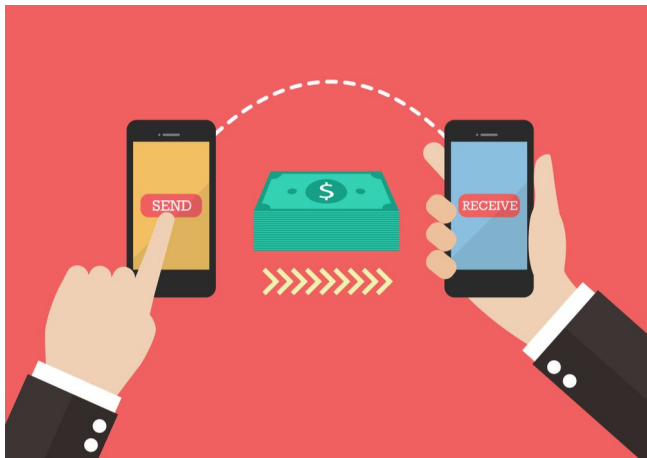
All the integrity constraints, rules that ensure validity and consistency of data, are maintained before and after a transaction.

Isolation

A transaction will not impact other transactions happening simultaneously in the database.

Durability

If a transaction is completed, then the changes made to the database due to that transaction remain there even if a database failure occurs.



What is the Transaction Control Language (TCL)?

TCL is a category of SQL language that is used to manage transactions inside a database.

The following commands are used in transaction control language:

1. COMMIT
2. ROLLBACK
3. SAVEPOINT



The **COMMIT** command is:

1. Used with DML
2. A transactional command and is used to save the changes made to a database
3. Used to saves all the transactions since the last **COMMIT** or **ROLLBACK** command

The syntax for the **COMMIT** command is as follows:
`COMMIT;`

Let's take a look at Example 6.



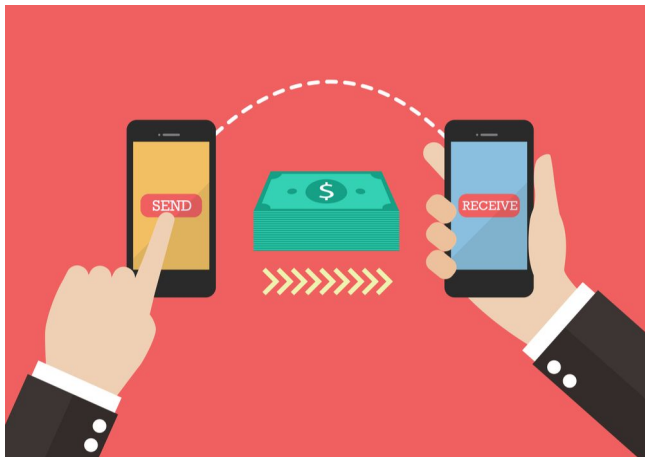
The ROLLBACK command is:

1. Used with DML
2. A transactional command and is used to undo the transactions that have not been saved already to a database
3. Used to undo all the transactions since the last **COMMIT** or **ROLLBACK** command

The syntax for the ROLLBACK command is as follows:

ROLLBACK;

Let's take a look at Example 7.



The SAVEPOINT command is:

1. Used with DML
2. Is a transactional command and is used to retrieve the transaction state to a certain point, instead of rolling back a complete transaction

The syntax for the SAVEPOINT command is as follows:

SAVEPOINT *savepoint_name*;

ROLLBACK TO *savepoint_name*;

Let's take a look at Example 8.

Poll 5

Which of the following commands can be used to save the temporary changes made to a database?

1. SAVE
2. COMMIT
3. PERSIST
4. None of the above

Poll 5 (Answer)

Which of the following commands can be used to save the temporary changes made to a database?

1. SAVE

2. COMMIT

3. PERSIST

4. None of the above

Poll 6

Which of the following commands can be rolled back?

1. DROP
2. DELETE
3. TRUNCATE

Poll 6

Which of the following commands can be rolled back?

1. DROP
2. **DELETE**
3. TRUNCATE



Constraints

1. They are rules enforced to ensure data integrity.
2. They help in avoiding data anomalies.

Constraints are of the following types:

NOT NULL

DEFAULT

CHECK

UNIQUE (Constraint as well as Key)

KEYS - Super Key, Candidate Key, Primary Key, Alternate/Secondary Key, Foreign Key, Natural Key, Surrogate Key



NOT NULL

It ensures that a column always has a non-NULL value.

To apply a NOT NULL constraint during table creation, use the following syntax:

```
CREATE TABLE table_name
(
    column1    Datatype,
    column2    Datatype,
    column3    Datatype NOT NULL,
    column4    Datatype NOT NULL
);
```

To apply a NOT NULL constraint during table alteration, use the following syntax:

```
ALTER TABLE table_name
MODIFY column_name Datatype NOT NULL;
```

Let's take a look at Example 9.

NOT NULL Command Hands-On Exercise (2 min)

Perform the following operations on the 'movie' table:

1. Apply the NOT NULL constraint to the following columns during table creation:
movie_name
movie_desc
release_date
2. Apply the NOT NULL constraint to the following columns during table alteration:
duration
cover_photo_url
trailer_url
language



DEFAULT

It ensures that a column always has a default value.

To apply a DEFAULT constraint during table creation, use the following syntax:

```
CREATE TABLE table_name
(
    column1    Datatype,
    column2    Datatype,
    column3    Datatype DEFAULT 'Value'
);
```

To apply a DEFAULT constraint during table alteration, use the following syntax:

```
ALTER TABLE table_name
MODIFY column_Name DEFAULT 'Value';
```

Let's take a look at Example 10.

DEFAULT Command Hands-On Exercise (2 min)

Perform the following operations on the 'theatre' table:

1. Set the DEFAULT value of *number_of_seats* to 50 during table creation.
2. Set the DEFAULT value of *ticket_price* to 150.00 during table alteration.



CHECK

It is used to limit the values that can be entered into a column.

It is applied at the following two levels:

1. **Single-column level:** Ensures that only limited values are allowed for that column.
2. **Table level:** Limits the values that can be entered into certain columns based on the values in other columns.

Syntax for CHECK constraint on a single column:

```
CREATE TABLE table_name
(
    column1    Datatype,
    column2    Datatype,
    column3    Datatype CHECK (condition)
);
```




Syntax for CHECK constraint on a table level:

```
CREATE TABLE table_name
(
    column1    Datatype,
    column2    Datatype,
    column3    Datatype
    CONSTRAINT constraint_name CHECK
    (condition)
);
```

Let's take a look at Example 11.

CHECK Command Hands-On Exercise (2 min)

In the 'movie' table, add a CHECK constraint such that movie_duration is always greater than 60.





UNIQUE

It ensures that all the values in a column are different.

During table creation use the following syntax:

```
CREATE TABLE table_name
(
    column1    Datatype,
    column2    Datatype,
    column3    Datatype UNIQUE
);
```

During table creation use the following syntax:

```
ALTER TABLE table_name
ADD UNIQUE (column_name) ;
```

Let's take a look at Example 12.

UNIQUE Command Hands-On Exercise (2 min)

In the movie table add a constraint such that two movies cannot have the same name.






KEYS are used for unique identification of records in a table. They are also used to establish relationships between tables

Keys are of the following types:

1. Super Key
2. Candidate key
3. Primary key
4. Alternate/secondary key
5. Foreign key
6. Natural key
7. Surrogate key

KEYS: Super Key

Super key is a group of single or multiple columns (or attributes) that uniquely identify the rows in a table.



	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEF...	❖ COLUMN...	❖ COMMEN...
1	FIRST_NAME	VARCHAR2(20 BYTE)	No	(null)	1	(null)
2	LAST_NAME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	USERNAME	VARCHAR2(20 BYTE)	No	(null)	3	(null)
4	PASSWORD	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5	MOBILE_NUMBER	CHAR(10 BYTE)	No	(null)	5	(null)
6	DATE_OF_BIRTH	DATE	No	(null)	6	(null)
7	USER_TYPE	VARCHAR2(20 BYTE)	Yes	'User'	7	(null)

Here are a few examples of super keys:

< first_name, last_name, username, password, mobile_number >

<first_name, last_name, username>

<first_name, mobile_number, username>



KEYS: Candidate Keys

Candidate keys are attributes or combinations of attributes that can uniquely identify each row of a table. Such combinations include only useful attributes. A candidate key is the minimal super keys.

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEF...	❖ COLUMN...	❖ COMMEN...
1	FIRST_NAME	VARCHAR2(20 BYTE)	No	(null)	1 (null)	
2	LAST_NAME	VARCHAR2(20 BYTE)	Yes	(null)	2 (null)	
3	USERNAME	VARCHAR2(20 BYTE)	No	(null)	3 (null)	
4	PASSWORD	VARCHAR2(20 BYTE)	No	(null)	4 (null)	
5	MOBILE_NUMBER	CHAR(10 BYTE)	No	(null)	5 (null)	
6	DATE_OF_BIRTH	DATE	No	(null)	6 (null)	
7	USER_TYPE	VARCHAR2(20 BYTE)	Yes	'User'	7 (null)	

Here is an example of a candidate key:

<username>



KEYS: Primary Keys

Primary keys are attributes or combinations of attributes that can uniquely identify each row of a table. Any candidate key can be chosen as a primary key.

Here are a few examples of candidate keys:

<username>
<customer_id>

- *There is only one primary key in a table.*

Name	Null?	Type
CUSTOMER_ID		CHAR (5)
FIRST_NAME		VARCHAR2 (20)
LAST_NAME		VARCHAR2 (20)
USERNAME		VARCHAR2 (20)
PASSWORD	NOT NULL	VARCHAR2 (20)
MOBILE_NO	NOT NULL	CHAR (10)
DATE_OF_BIRTH	NOT NULL	DATE
USER_TYPE		VARCHAR2 (20)

Let's take a look at Example 13.

Primary Key Hands-On Exercise (2 min)

Add a primary key to the 'movies' table.





KEYS: Alternate Keys

Alternate keys are attributes or combinations of attributes that can uniquely identify each row of a table. An alternate keys are all the candidate keys except the one that is chosen as the primary key.

	⚡ COLUMN_NAME	⚡ DATA_TYPE	⚡ NULLABLE	DATA_DEF...	⚡ COLUMN...	⚡ COMMEN...
1	CUSTOMER_ID	CHAR(5 BYTE)	No	(null)	1	(null)
2	FIRST_NAME	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	LAST_NAME	VARCHAR2(20 BYTE)	Yes	(null)	3	(null)
4	USERNAME	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5	PASSWORD	VARCHAR2(20 BYTE)	No	(null)	5	(null)
6	MOBILE_NUMBER	CHAR(10 BYTE)	No	(null)	6	(null)
7	DATE_OF_BIRTH	DATE	No	(null)	7	(null)
8	USER_TYPE	VARCHAR2(20 BYTE)	Yes	'User' ...	8	(null)

Here is an example of an alternate key:

<username> *is an alternate key*
<customer_ID> (primary key)



KEYS: Foreign Keys

A Foreign Key is a column that points to the Primary Key of other table and maintain the relationship between two tables. It is also used to ensure referential integrity.

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	CUSTOMER_ID	CHAR(5 BYTE)	No	(null)	1 (null)	
2	FIRST_NAME	VARCHAR2(20 BYTE)	No	(null)	2 (null)	
3	LAST_NAME	VARCHAR2(20 BYTE)	Yes	(null)	3 (null)	
4	USERNAME	VARCHAR2(20 BYTE)	No	(null)	4 (null)	
5	PASSWORD	VARCHAR2(20 BYTE)	No	(null)	5 (null)	
6	MOBILE_NUMBER	CHAR(10 BYTE)	No	(null)	6 (null)	
7	DATE_OF_BIRTH	DATE	No	(null)	7 (null)	
8	USER_TYPE_ID	CHAR(5 BYTE)	Yes	(null)	8 (null)	

user_type_id, which is a foreign key in the **customer** table, is used to reference the user_type_id column (Primary Key) in the **user_type** table.

Let's take a look at Example 14.

FOREIGN KEY Command Hands-On Exercise (5 min)

Every movie has a corresponding status, which is stored in a separate table named 'status', which has the following structure:

status_id CHAR(5) - PRIMARY KEY

status_name VARCHAR(20) - cannot be NULL and cannot store duplicate values

Create the 'status' table and link the status table with the 'movie' table.



KEYS: Natural and Surrogate Keys

Natural Keys

Natural keys are the keys that hold business meaning. They use one of the key attributes of the entity to identify each record uniquely.

For example, using ***user_name*** as a key to identify each record in a ***user*** table.

Surrogate Keys

Surrogate keys have no business meaning. The sole purpose of using them is to identify each record in a table uniquely. They need to be explicitly to the table.

For example, using ***id*** as a key to identify each record in a table.

Poll 7

Which of the following is an example of super key?

1. CANDIDATE
2. PRIMARY KEY
3. BOTH
4. NONE

Poll 7 (Answer)

Which of the following is a Super Key with minimal necessary attributes?

1. CANDIDATE
2. PRIMARY KEY
- 3. BOTH**
4. NONE

Poll 8

Any attribute which is not a Primary key, in a table can be added as a Foreign Key to another table?

1. TRUE

2. FALSE

Poll 8 (Answer)

Any attribute which is not a Primary key, in a table can be added as a Foreign Key to another table?

1. TRUE

2. **FALSE**

Poll 9

What is an alternate key?

1. All the candidates keys
2. A Candidate key that is not chosen as the primary key
3. Primary key
4. None of the above

Poll 9 (Answer)

What is an alternate key?

1. All the candidates keys

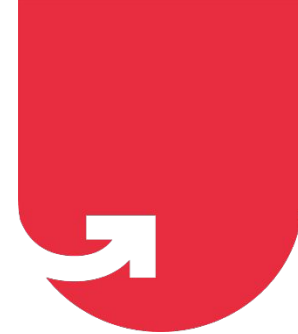
2. A Candidate key that is not chosen as the primary key

3. Primary key

4. None of the above



DISCUSSION TIME



Thank You!