



XILYTICA - DATABASE & SQL

Introduction

- **Installing MySQL**
- **Installing Workbench**

<https://dev.mysql.com/downloads/mysql/>

<https://downloads.mysql.com/archives/workbench/>

Introduction

- Databases
- Database Management System
- SQL

Introduction

- We will download and work on large database that is employee database from GitHub ,to understand how SQL is used in real life
- How to query & extract information & insights from the database
- We look into -
 - SQL Statements
 - Column & Table Constraints
 - Built-In Functions
 - Joins
 - Sub Queries
 - Views
 - Functions
 - Stored Procedures
 - Indexes
 - Triggers

SQL

- SQL - Structured Query Language
- Specifically designed for working with databases specifically **relational database system** (RDBMS)
- It is used for
 - Create
 - Manipulate
 - Share
- Query - Piece of code when executed on database, returns desired output
- SQL, allows us to write query that executes on database & returns outputs in the form of data
- SQL is vastly used by
 - Web Programmer
 - Data Analyst
 - Business Intelligence Analyst
 - Data Scientist
- SQL is one of the most in demand programming language as per the [berkeley.edu](https://bootcamp.berkeley.edu/blog/most-in-demand-programming-languages)

Database Management System

- Databases in the market based on SQL with slight differences
 - MySQL (Our Choice) - Open Source & Free
 - PostgreSQL
 - Oracle
 - Microsoft SQL Server
 - Microsoft Access
 - SQL Lite
 - Maria DB



MySQL

MySQL is the world's most popular open source database



LinkedIn

YouTube

Dropbox



NETFLIX

slack

airbnb



Uber

Databases

DBMS - Database Management System as a "software system that enables users to define, create, maintain and control access to the database

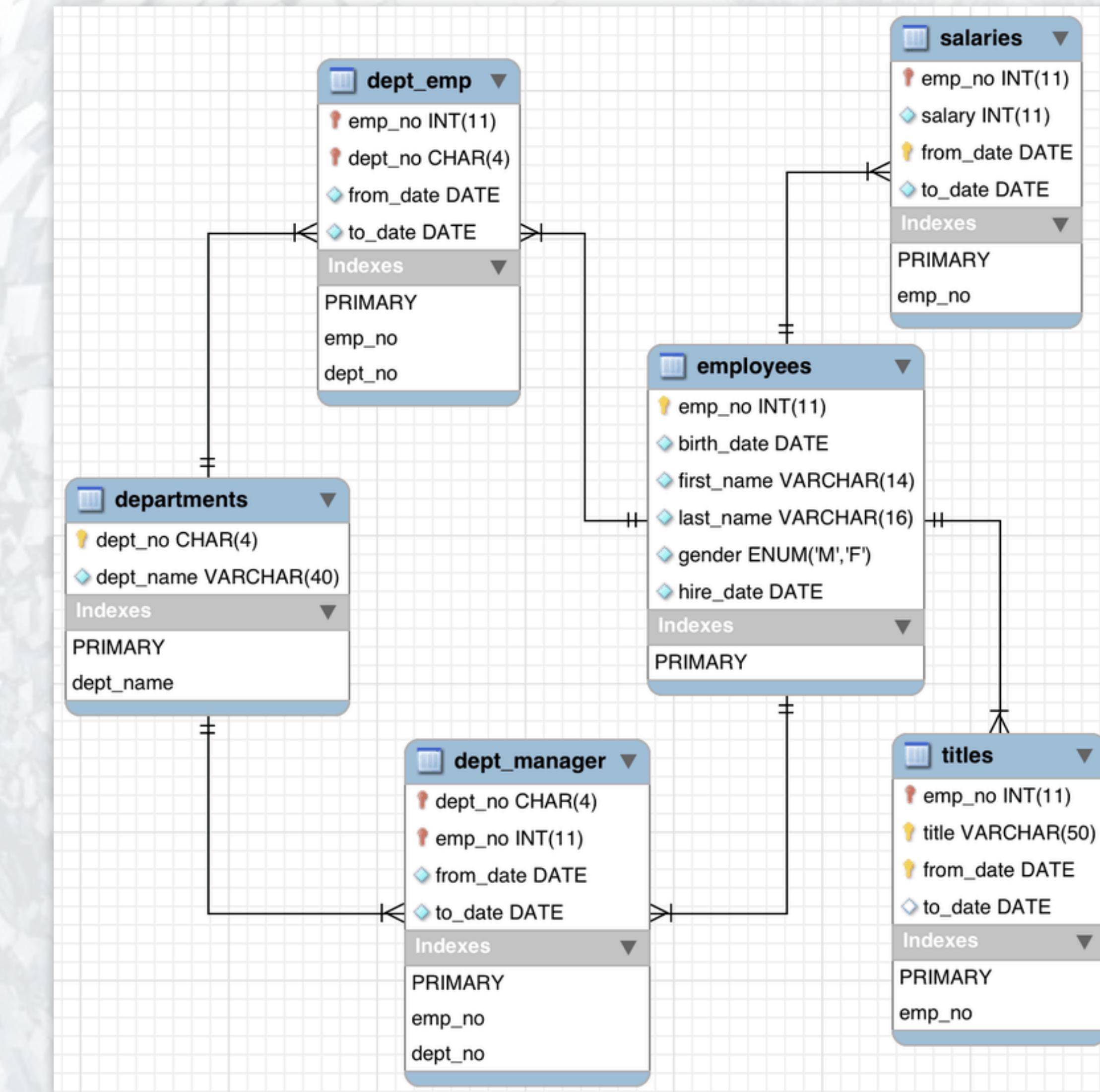
RDBMS - A relational database Management System is a type of database that stores and provides access to data points that are related to one another. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

In RDBMS -

- Data is stored in tabular form - Rows & Columns
- Entity - Smallest unit that can contain a meaning set of data
 - Row are horizontal entity
 - Columns are vertical entity
 - Table is also an entity as well as database object
 - Row of a table can be called as entity instance

Employee database Data Flow Diagram

- ER Model - Entity Relationship Model - Describes the structure of database with the help of a diagram which is known as ER Diagram.



<https://www.lucidchart.com/pages/er-diagrams>

<https://dev.mysql.com/doc/employee/en/sakila-structure.html>

SQL Syntax

SQL - To create & manipulate relational database. It's a declarative programming language. We are more interested in what should be an output.

Main components of a SQL Syntax are

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Transactional Control Language (TCL)

Data Definition Language - DDL

A set of statements that allow user to define or modify data structures & objects, such as table

Allows us to -

- CREATE
- ALTER
- RENAME
- DROP
- TRUNCATE

Data Definition Language - DDL

CREATE STATEMENT -

Used for creating entire database or database objects such as tables

```
CREATE TABLE products  
(sku int, product text, name varchar(14), product_id int, qty int, price float);
```

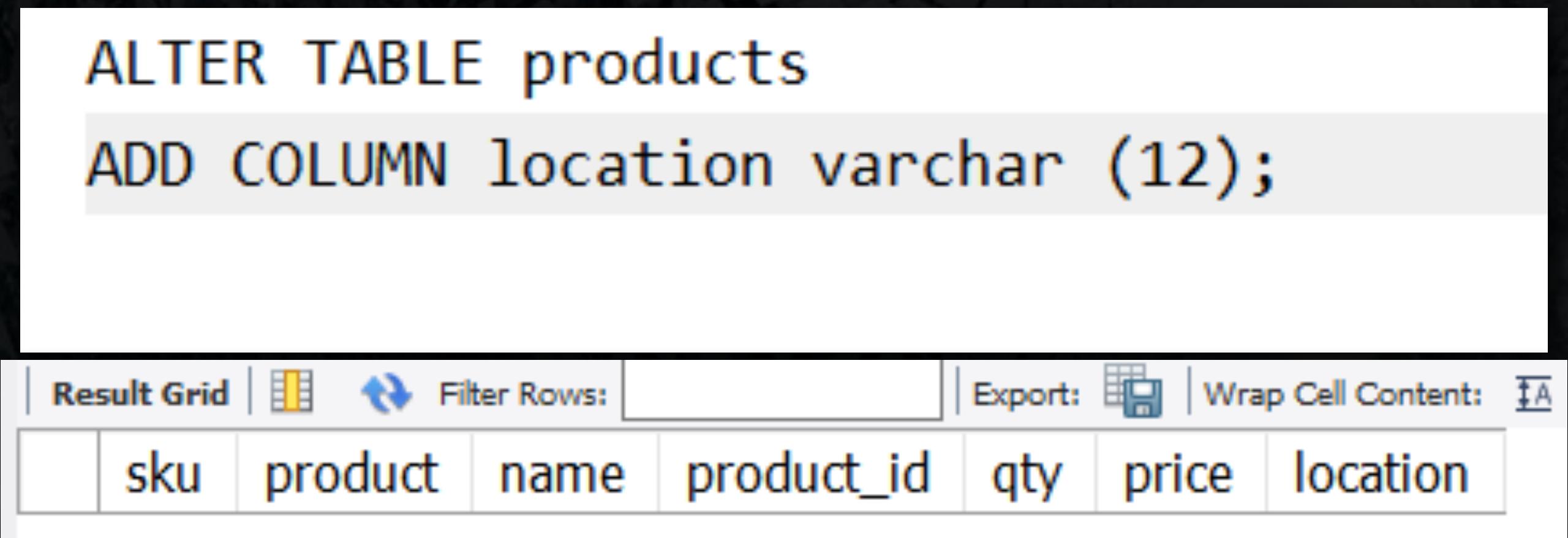
Data Definition Language - DDL

Alter Statement -

Used when altering existing objects

- ADD - Will add a database object
- RENAME
- DROP

```
ALTER TABLE products
ADD COLUMN location varchar (12);
```



The screenshot shows a MySQL Workbench interface. At the top, there is a code editor window containing the SQL command: `ALTER TABLE products ADD COLUMN location varchar (12);`. Below the code editor is a results grid. The grid has a header row with columns labeled `sku`, `product`, `name`, `product_id`, `qty`, `price`, and `location`. The grid itself is currently empty, with no data rows visible.

Data Definition Language - DDL

Alter Statement -

Used when altering existing objects

- ADD
- RENAME - Will rename the database object
- DROP

```
RENAME TABLE products to my_products;
```

Data Definition Language - DDL

Alter Statement -

Used when altering existing objects

- ADD
- RENAME
- DROP - Will drop the object from the database

```
• DROP TABLE products;
```

Data Definition Language - DDL

Truncate - Deleting entire data from the table in one shot Very Fast. Table remains but data is deleted

- TRUNCATE TABLE products;

Data Manipulation Language - DML

It allows us to manipulate the data in the tables of a database

- SELECT Statement
- INSERT Statement
- UPDATE Statement
- DELETE Statement

SELECTFROM
INSERT INTOVALUES
UPDATESET.....WHERE
DELETE FROM WHERE

Data Manipulation Language - DML

SELECT Statement - Used to retrieve data from database objects like tables

```
4
5 • select * from employees;
```

The screenshot shows a MySQL Workbench interface with a result grid. The grid has columns for emp_no, birth_date, first_name, last_name, gender, and hire_date. The data consists of eight rows, each representing an employee from the employees table. The grid includes standard database navigation and editing tools at the top.

	emp_no	birth_date	first_name	last_name	gender	hire_date
▶	10001	1953-09-02	Georgi	Facello	M	1986-06-26
	10002	1964-06-02	Bezalel	Simmel	F	1985-11-21
	10003	1959-12-03	Parto	Bamford	M	1986-08-28
	10004	1954-05-01	Chirstian	Koblick	M	1986-12-01
	10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12
	10006	1953-04-20	Anneke	Preusig	F	1989-06-02
	10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
	10008	1958-02-19	Saniya	Kalloufi	M	1994-09-15

Data Manipulation Language - DML

INSERT Statement - Used to insert data into the tables

```
2
3 • insert into products (sku, product, name, product_id, qty, price, location)
4   VALUES ( 100, 'iphone', 'mobile', 100, 50, 7000, 'NY');
5
6 • insert into products
7   VALUES ( 101, 'galaxy Note', 'mobile', 101, 30, 5000, 'NJ');
```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The results grid displays the data inserted into the 'products' table:

	sku	product	name	product_id	qty	price	location
▶	100	iphone	mobile	100	50	7000	NY
	101	galaxy Note	mobile	101	30	5000	NJ

Data Manipulation Language - DML

UPDATE Statement - Used to update existing record of the table

```
10 • UPDATE PRODUCTS
11   SET name = 'Samsung Mobile'
12 WHERE sku = 101;
13
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	sku	product	name	product_id	qty	price	location
▶	100	iphone	mobile	100	50	7000	NY
	101	galaxy Note	Samsung Mobile	101	30	5000	NJ

Data Manipulation Language - DML

DELETE Statement - Delete the record based on your where clause that is which recode to delete

```
14  
15  /*# Will delete all records from the table */  
16 • DELETE FROM products ;  
17  
18  /*Will deleted recodes based on where clause */  
19 • DELETE FROM products  
20 WHERE sku = 101;  
21
```

Data Control Language - DCL

It allows us to manage the rights users have in the database. Databases is used by n number of users where n could be 10, or 100 or 1000 or even more. But not all have same right to access the information in the database.

- GRANT Statement
- REVOKE Statement

Data Control Language - DCL

GRANT Statement - Gives or grants certain permission to the users.

```
22  
23 # Creating a user test_user with password as password  
24 • CREATE USER 'test_user'@'localhost' IDENTIFIED BY 'password';  
25  
26 • GRANT SELECT ON employees.products TO 'test_user'@'localhost';  
27  
28 • SELECT * FROM products;  
29
```

Data Control Language - DCL

GRANT Statement - Gives or grants certain permission to the users.

privileges

It can be any of the following values:

Privilege	Description
SELECT	Ability to perform SELECT statements on the table.
INSERT	Ability to perform INSERT statements on the table.
UPDATE	Ability to perform UPDATE statements on the table.
DELETE	Ability to perform DELETE statements on the table.
INDEX	Ability to create an index on an existing table.
CREATE	Ability to perform CREATE TABLE statements.
ALTER	Ability to perform ALTER TABLE statements to change the table definition.
DROP	Ability to perform DROP TABLE statements.
GRANT OPTION	Allows you to grant the privileges that you possess to other users.
ALL	Grants all permissions except GRANT OPTION.

Data Control Language - DCL

GRANT Statement - Gives or grants certain permission to the users.

The screenshot shows a database management system interface with the following components:

- Navigator:** Shows the schema structure. Under the 'employees' schema, there is a 'products' table. Other objects like Views, Stored Procedures, and Functions are also listed.
- Query 1:** A query window containing the SQL command: `1 • SELECT * FROM products;`
- Result Grid:** A table displaying the results of the query:

sku	product	name	product_id	qty	price	location
100	iphone	Apple	100	50	7000	NY
- SQLAdditions:** A panel on the right with the message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."
- Output:** A log window showing the execution details:

#	Time	Action
1	07:08:08	SELECT * FROM products LIMIT 0, 1000

Message: 1 row(s) returned

Data Control Language - DCL

If only SELECT permission is given to the user than the user will not be able to perform other operations on the database such as DROP, TRUNCATE, ALTER, INSERT, DELETE

The screenshot shows the MySQL Workbench interface. On the left, the schema browser displays the 'employees' database with its tables ('products'), columns, indexes, foreign keys, and triggers. The 'Actions' tab is selected. In the main pane, a query editor contains the following code:

```
2
3
4 • UPDATE products
5 SET name='SAMSUNG'
6 WHERE sku = 100;
```

Below the code, the 'Output' tab is open, showing the results of the query. A single row is listed:

#	Time	Action	Message
1	07:11:49	UPDATE products SET name='SAMSUNG' WHERE sku = 100	Error Code: 1142. UPDATE command denied to user 'test_user'@'localhost' for table 'products'

The message indicates that the user 'test_user' does not have the necessary permissions to execute the UPDATE command on the 'products' table.

Data Control Language - DCL

Database Administrator - To give all permission to the user we have to login as a administrator and provide GRANT ALL to the user. After this user can perform all DDL & DML operations on the database. Also all the tables will be shown on user's profile

The screenshot shows a MySQL Workbench interface. The SQL editor contains the following code:

```
1 # To grant all operations
2
3 • GRANT ALL ON employees.* to 'test_user'@'localhost';
```

The output pane shows the results of the query:

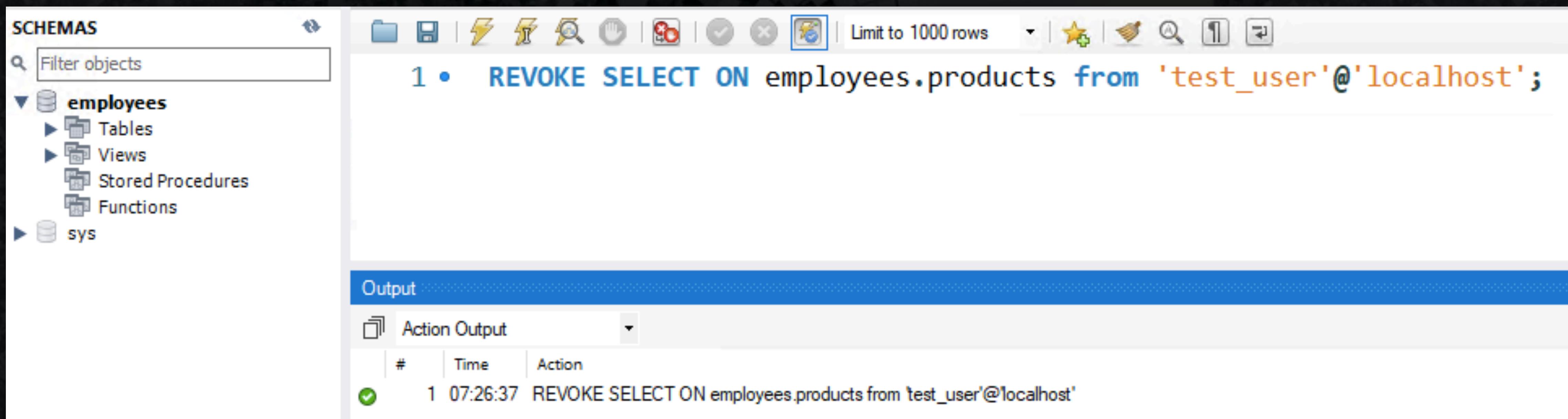
Action Output
Time Action
1 07:18:09 GRANT ALL ON employees.* to 'test_user'@'localhost'

Message: 0 row(s) affected

Data Control Language - DCL

REVOKE Statement - Used to revoke permissions & privileges of database users.

REVOKE type_of_permission on database_name.table_name from 'user'@'localhost'



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing the 'employees' schema expanded to show Tables, Views, Stored Procedures, and Functions. Below it is the 'sys' schema. The main pane contains a SQL editor with the following query:

```
1 • REVOKE SELECT ON employees.products FROM 'test_user'@'localhost';
```

The status bar at the bottom of the editor indicates the query was executed successfully. The 'Output' tab is selected, showing the 'Action Output' section with a single row of results:

#	Time	Action
1	07:26:37	REVOKE SELECT ON employees.products FROM 'test_user'@'localhost'

Data Control Language - DCL

REVOKE Statement - Used to revoke permissions & privileges of database users.

REVOKE type_of_permission on database_name.table_name from 'user'@'localhost'

privileges

It can be any of the following values:

Privilege	Description
SELECT	Ability to perform SELECT statements on the table.
INSERT	Ability to perform INSERT statements on the table.
UPDATE	Ability to perform UPDATE statements on the table.
DELETE	Ability to perform DELETE statements on the table.
INDEX	Ability to create an index on an existing table.
CREATE	Ability to perform CREATE TABLE statements.
ALTER	Ability to perform ALTER TABLE statements to change the table definition.
DROP	Ability to perform DROP TABLE statements.
GRANT OPTION	Allows you to grant the privileges that you possess to other users.
ALL	Grants all permissions except GRANT OPTION.

Transactional Control Language - TCL

Changes that you made to the database are saved automatically.

- COMMIT Statement
- ROLLBACK Statement

```
# Will switch off auto commit
• SET SESSION autocommit = 0;
-- or
• SET SESSION autocommit = OFF;

# Will switch on auto commit
• SET SESSION autocommit = 1;
-- or
• SET SESSION autocommit = ON;
```

Transactional Control Language - TCL

COMMIT Statement

- Save the changes that you made on the database permanent.
- It works on DML operations such as INSERT, DELETE, UPDATE
- Once you commit, then other users will have access to the updated database

Transactional Control Language - TCL

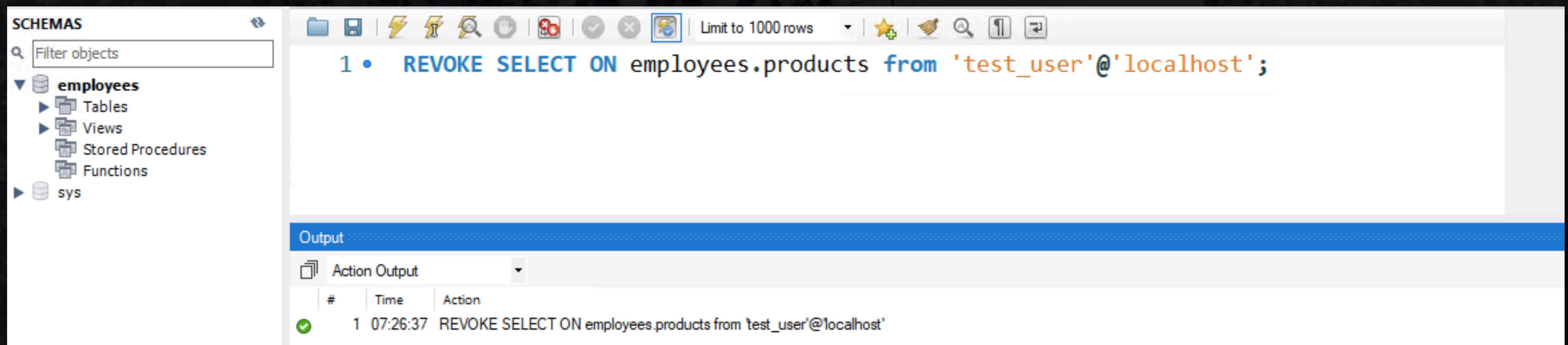
ROLLBACK Statement

- It will rollback the current changes

Data Control Language - DCL

REVOKE - Used to revoke permissions & privileges of database users.

REVOKE type_of_permission on database_name.table_name from 'user'@'localhost'



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing the 'employees' schema expanded to show Tables, Views, Stored Procedures, and Functions. Below it is the 'sys' schema. In the center, the SQL editor window contains the following command:

```
1 • REVOKE SELECT ON employees.products FROM 'test_user'@'localhost';
```

Below the editor is the 'Output' tab, which displays the execution results:

Action Output		
#	Time	Action
1	07:26:37	REVOKE SELECT ON employees.products FROM 'test_user'@'localhost'

The output row has a green checkmark next to it, indicating the command was successful.

Data Definition Language - DDL

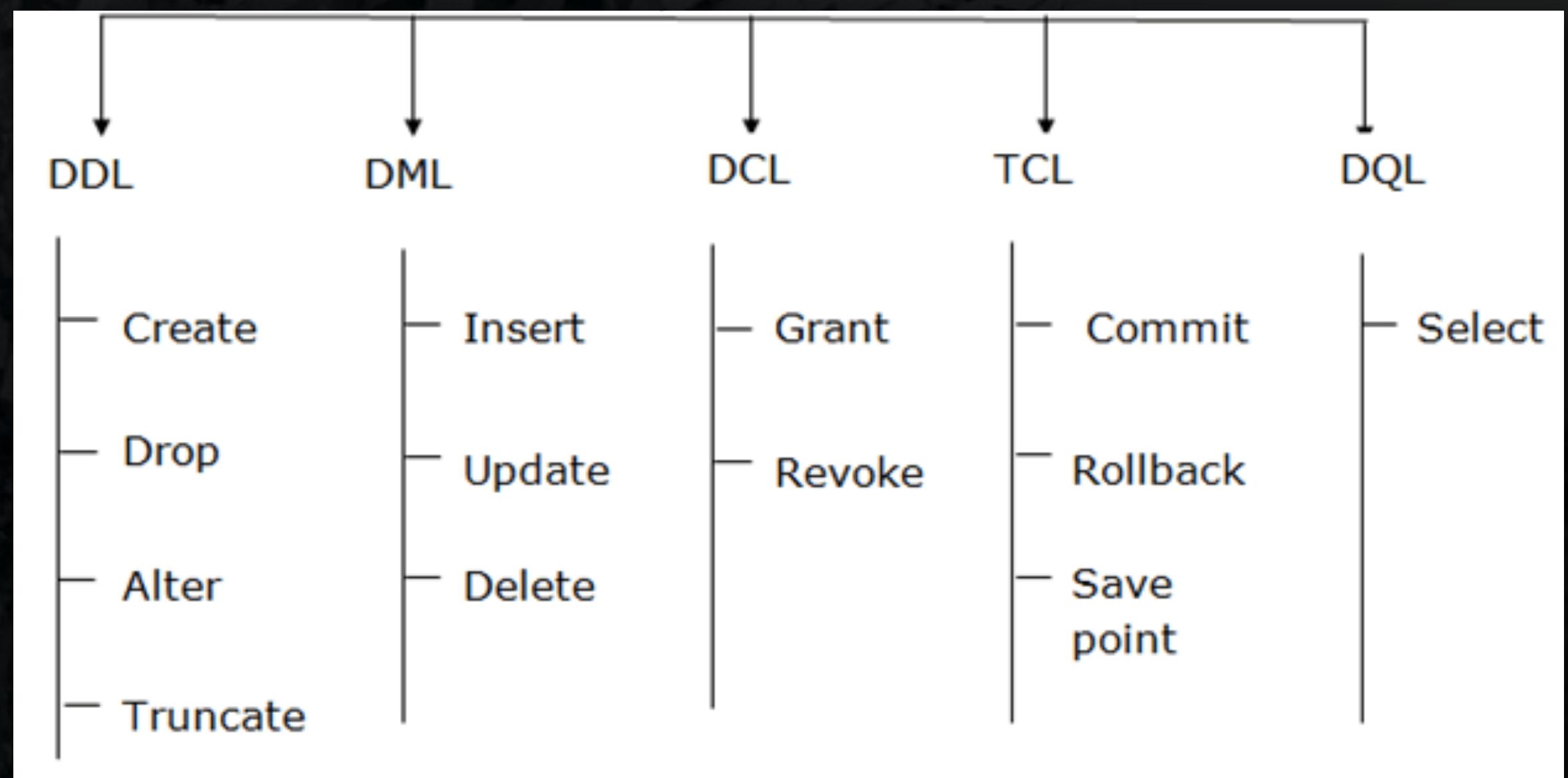
A set of statements that allow user to define or modify data structures & objects, such as table

Create Statement -

Used for creating entire database or database objects such as tables

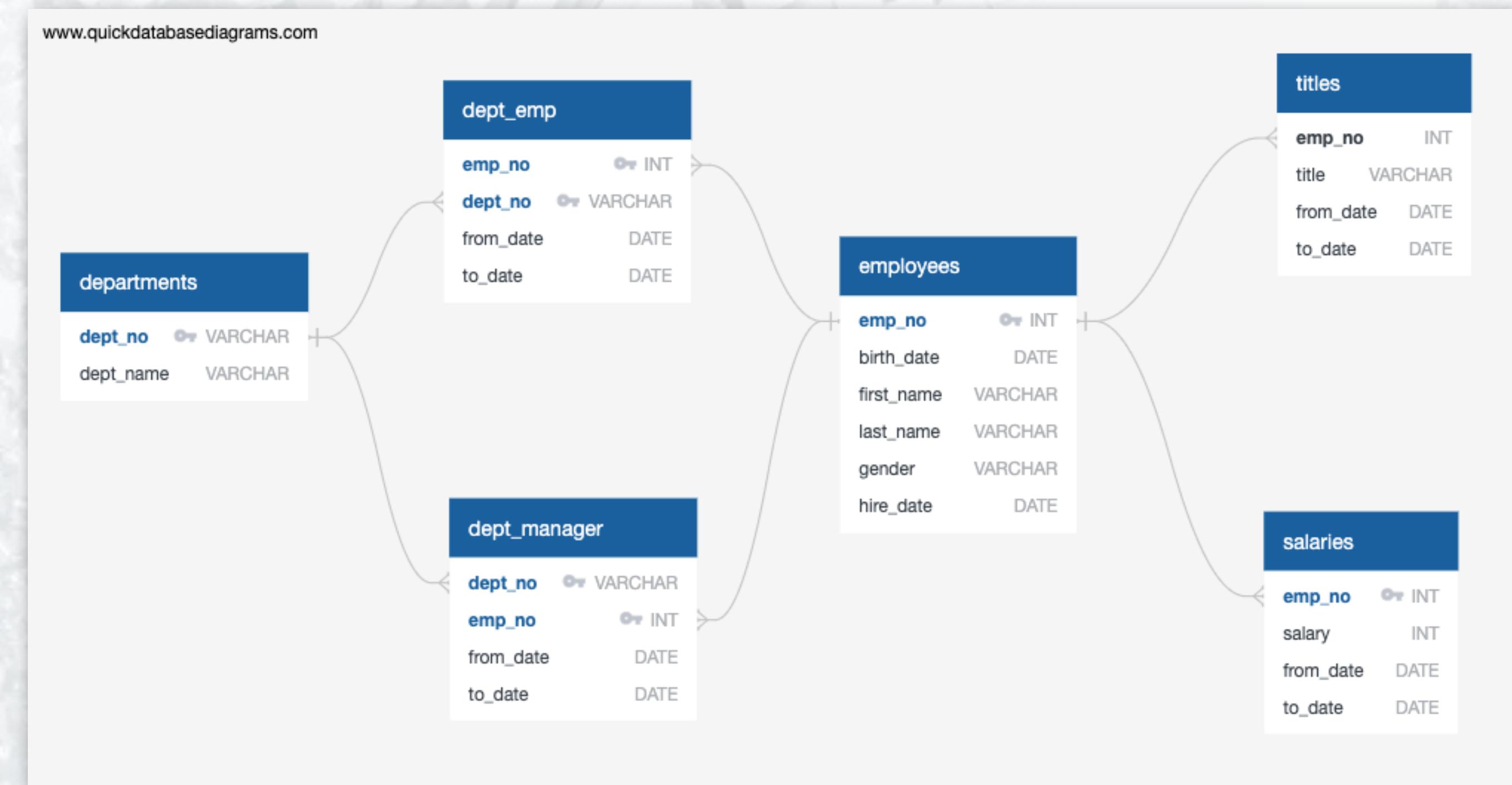
- `CREATE TABLE products
(sku int, product text, name varchar(14), product_id int, qty int, price float);`
- `Select * from products;`

SQL



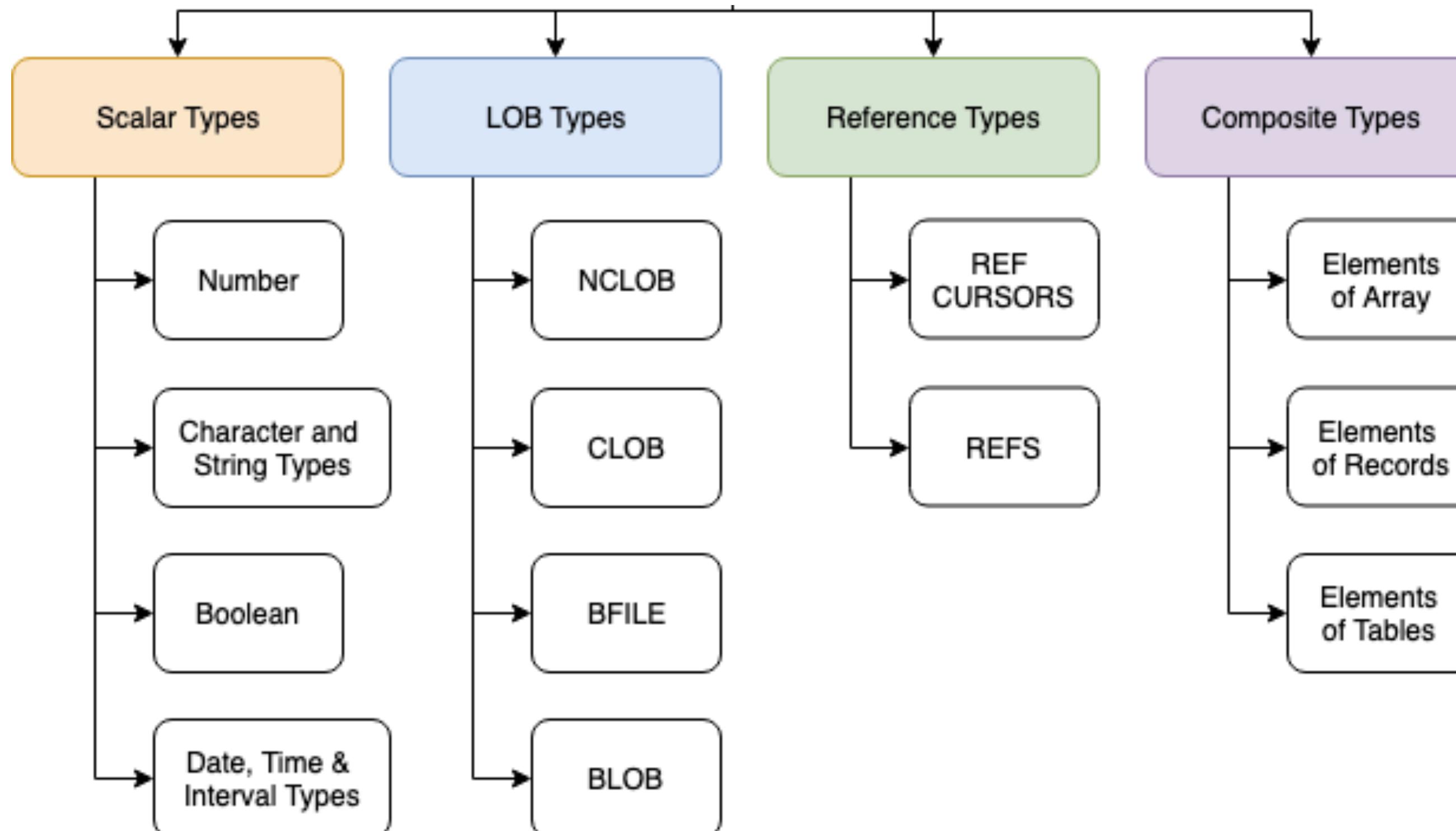
Databases

- Data is stored in tabular form - Rows & Columns

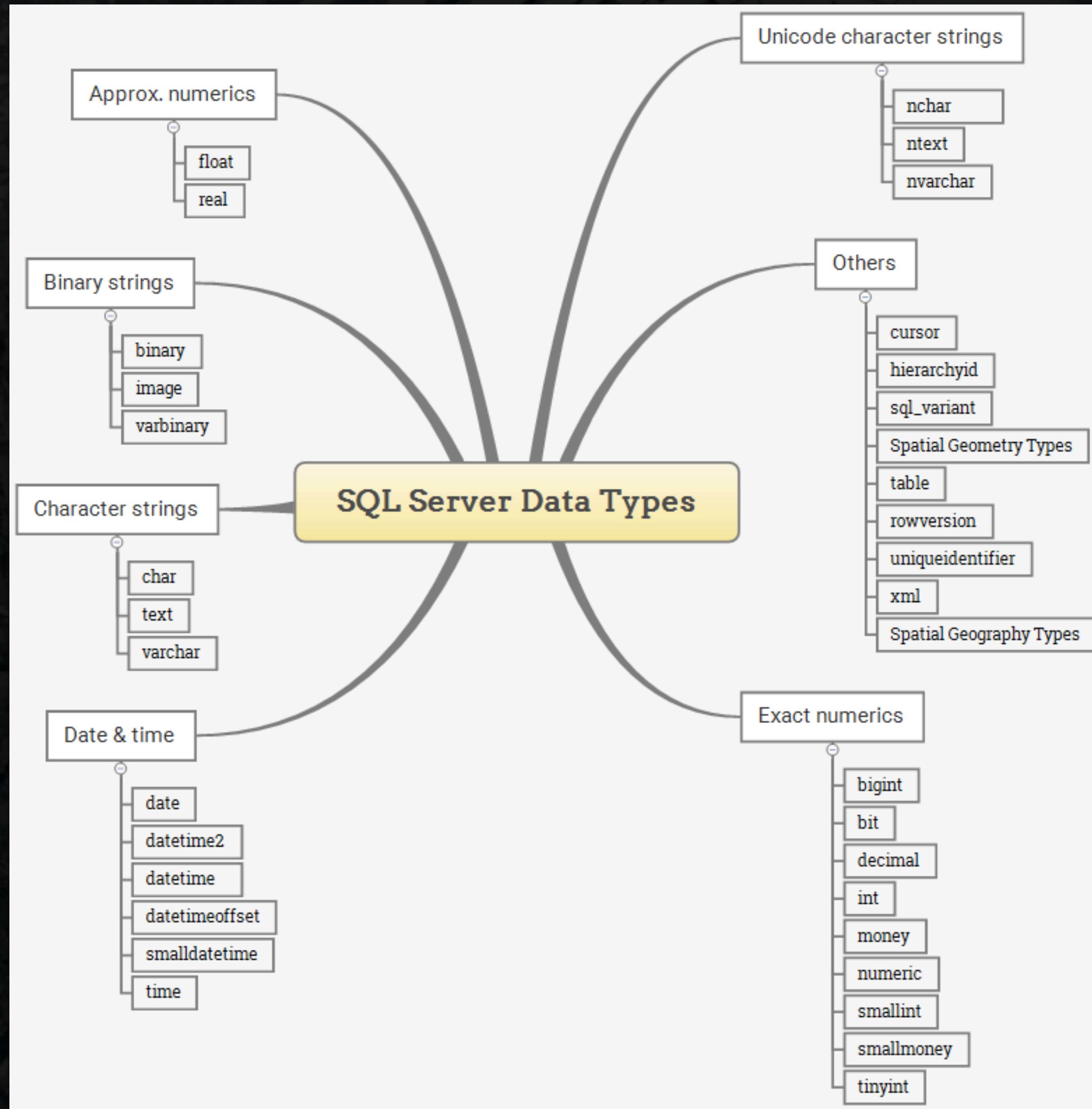


Creating Database or Schema

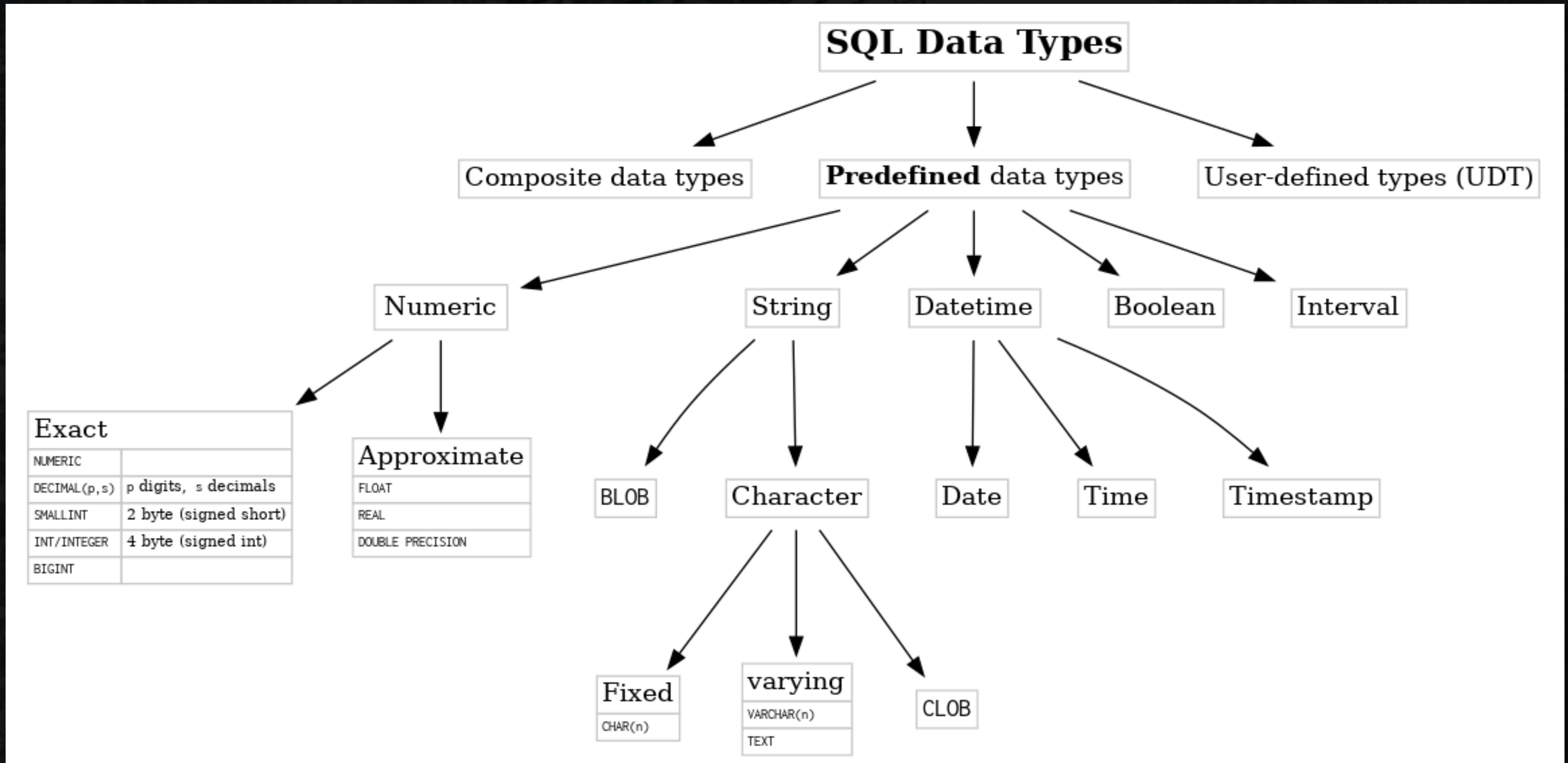
Data Types in SQL



Data Types in SQL



Data Types in SQL



Data Types in SQL

String - Text or Alphanumeric type data

String Data Type	Data Type In SQL	Storage	Example	Length	Byte Size Allotted	Maximum Size (Bytes)
Character	CHAR	Fixed	CHAR(5) 'Nitin' 'Bill'	5 4	5 5	255
Variable Character	VARCHAR	Variable	VARCHAR(5)	5 4	5 4	65,535
ENUM Enumerate	ENUM		ENUM('M','F') This will take only M or F			

Number in brackets () - Maximum length that variable can take.

CHAR is 50% faster compared to VARCHAR

Data Types in SQL

Number - Numeric data type

Integer

- ✓ Integer

Fixed Integer

- ✓ Decimal
- ✓ Numeric

Floating Point

- ✓ Float
- ✓ Double

Data Types in SQL

Integer (INT) - Whole numbers with no decimals. Eg. 2,15, 20, -100

Type	Storage (Bytes)	Minimum Value Signed	Maximum Value Signed	Minimum Value Unsigned	Maximum Value Unsigned
TINYINT	1	-128	127	0	255
SMALLINT	2	-32768	32767	0	65535
MEDIUMINT	3	-8388608	8388607	0	16777215
INT	4	-2147483648	2147483647	0	4294967295
BIGINT	8	-9.22337E+18	2^63-1	0	2^64-1

Signed - Range includes both positive & negative values. Default is Signed

Unsigned - Range includes only positive values

Always use right data type for better performance & efficiency

Data Types in SQL

Fixed Point - Represents exact value eg. decimal(5,2) for 123.45. If you try to insert number more than this, sql will give you a warning and will slice the number based on precision & Scale

Number	Precision	Scale	Example
123.45	5	2	Decimal(5,2)
12.4567	5	2	12.45 System will throw a warning

Precision - Total number of digits in a number

Scale - Number of digits after decimal point

Decimal = Numeric data type

Decimal(5,2) = Numeric(5,2)

Data Types in SQL

Floating Point - Used for approximate values only. Aims to balance between range & precision . If you try to insert number more than this, sql will not give any warning and round off the number

Number	Precision	Scale	Example
123.45	5	2	Decimal(5,2)
12.4567	5	2	12.46

Precision - Total number of digits in a number

Scale - Number of digits after decimal point

Data Types in SQL

Floating Point - Used for approximate values only. Aims to balance between range & precision . If you try to insert number more than this, sql will not give any warning and round off the number

Floating Point Data Type	Size (Bytes)	Precision	Max. Number of Digits
Float	4	Single	23
Double	8	Double	53

Data Types in SQL

Fixed Point - Represents exact value eg. decimal(5,2) for 123.45. If you try to insert number more than this sql will give you a warning and will slice the number based on precision & Scale

Number	Precision	Scale	Example
123.45	5	2	Decimal(5,2)

Precision - Total number of digits in a number

Scale - Number of digits after decimal point

Decimal = Numeric data type

Decimal(5,2) = Numeric(5,2)

Date & Time

DATE	YYYY-MM-DD	
DATETIME	YYYY-MM-DD HH:MM:SS	Calendar Day & Time
TIMESTAMP	YYYY-MM-DD HH:MM:SS	Well defined, exact point in time

Date & Time

Similarity Between DATETIME & TIMESTAMP

- Both store data in "YYYY-MM-DD HH:MM:SS" format.
- Both include date as well as time part.
- Automatic initialization can happen for both.
- Both change the data while updating the record with current data time as per the constraint.
- Both can have fractional seconds part upto 6 digit microsecond precision.
-

Difference Between DATETIME & TIMESTAMP

- Supported range for
 - DATETIME is '1000-01-01 00:00:00' to '9999-12-31 23:59:59'
 - TIMESTAMP, it is '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC
- TIMESTAMP differs with current timezone settings while DATETIME remains constant.
-

Date & Time

DATETIME Vs TIMESTAMP

DATETIME	TIMESTAMP
Calendar time	Records the moment in time as the number of seconds passed after 1st January 1970 00:00:00 UTC
Supported range - '1000-01-01 00:00:00' to '9999-12-31 23:59:59'	Supported range - '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC
Current timezone settings remain same	Current timezone setting differs

With or Without Quotes

WITH QUOTES	WITHOUT QUOTES
CHAR	INTEGER
VARCHAR	DECIMAL
DATE	NUMERIC
DATETIME	FLOAT
TIMESTAMP	DOUBLE

BLOB

BLOB - Binary Large Object

- Files of binary (0,1) data
- Saving files in records
- Supported file types are - *.doc, *.xlxs, *.xml, *.jpg, *.wav

Creating a Table

```
# Creating a table with atleast one column is required
# Syntax

CREATE TABLE TABLE_NAME
(
    column_1    data_type    constraints,
    column_2    data_type    constraints,
    .
    .
    .
    column_n    data_type    constraints
);
```

Constraints

Constraints are specific rules or limits that we define on our tables based on our requirement, business or otherwise .

Relationships between different tables in our database

Few of the constraints are -

- NOT NULL
- PRIMARY KEY
- FOREIGN KEY
- UNIQUE KEY

Primary Key Constraints

Primary Key Constraint -

- The PRIMARY KEY constraint uniquely identifies each record in a table.
- Primary keys must contain UNIQUE values, and cannot contain NULL values.
- A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields)

```
CREATE TABLE sales
(
    purchase_number  INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    date_of_purchase DATE NOT NULL,
    customer_id INT,
    item_code VARCHAR(10) NOT NULL
);
```

```
CREATE TABLE sales
(
    purchase_number  INT NOT NULL AUTO_INCREMENT,
    date_of_purchase DATE NOT NULL,
    customer_id INT,
    item_code VARCHAR(10) NOT NULL,
    PRIMARY KEY (purchase_number)
);
```

Foreign Key Constraints

Foreign Key Constraint -

- A foreign key is a key used to link two tables together. This is sometimes also called as a referencing key.
- A Foreign Key is a column or a combination of columns whose values match a Primary Key in a different table.
- The relationship between 2 tables matches the Primary Key in one of the tables with a Foreign Key in the second table.
- The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

```
CREATE TABLE sales
(
    purchase_number INT NOT NULL AUTO_INCREMENT,
    date_of_purchase DATE NOT NULL,
    customer_id INT
    item_code VARCHAR(10),
    PRIMARY KEY (purchase_number),
    FOREIGN KEY (customer_id)
);
# Deleting foreign key
/*ALTER TABLE sales
DROP FOREIGN KEY sales_ibfk_1;
*/
ON DELETE CASCADE
```

```
ALTER TABLE sales
ADD FOREIGN KEY (customer_id) REFERENCES customers(customer_id) ON DELETE CASCADE
```

Foreign Key Constraints

Foreign Key Constraint -

```
CREATE TABLE sales
(
    purchase_number INT NOT NULL AUTO_INCREMENT,
    date_of_purchase DATE NOT NULL,
    customer_id INT,
    item_code VARCHAR(10) NOT NULL,
    PRIMARY KEY (purchase_number),
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id) ON DELETE CASCADE
);

ALTER TABLE sales
ADD FOREIGN KEY (customer_id) REFERENCES customers(customer_id) ON DELETE CASCADE
```

```
# Deleting foreign key
ALTER TABLE sales
DROP FOREIGN KEY sales_ibfk_1;
```

Foreign Key Constraints

ON DELETE CASCADE

It is used to automatically remove the matching records from the child table when we **delete** the rows from the parent table. It is a kind of referential action related to the foreign key.

Unique Key Constraints

Unique Key

- The UNIQUE constraint ensures that all values in a column are different.
- Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.
- A PRIMARY KEY constraint automatically has a UNIQUE constraint.
- However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

```
CREATE TABLE customers
(
    customer_id INT,
    first_name varchar(255),
    last_name varchar(255),
    email_address varchar(255),
    number_of_complaints int,
    primary key (customer_id) ,
    unique key (email_address)
);
```

```
ALTER TABLE customers
ADD UNIQUE KEY (email_address);
```

Indexes

- Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.
- Indexes are special lookup tables that the database search engine can use to speed up data retrieval. In simpler words, an index is a pointer to data in a table. An index in a database is very similar to an index in the back of a book.
- An index helps to speed up SELECT queries and WHERE clauses, but it slows down data input, with the UPDATE and the INSERT statements as Indexes also need an update.
- Indexes can be created or dropped with no effect on the data.

```
CREATE INDEX index_name  
ON table_name (column1, column2, ...);
```

```
CREATE UNIQUE INDEX index_name  
ON table_name (column1, column2, ...);
```

```
# Droping an Index  
ALTER TABLE customers  
DROP INDEX email_address;
```

Default Constraint

- The DEFAULT constraint is used to set a default value for a column.
- The default value will be added to all new records, if no other value is specified.

```
CREATE TABLE customers
(
    customer_id INT,
    first_name varchar(255),
    last_name varchar(255),
    email_address varchar(255),
    number_of_complaints int default 0,
    primary key (customer_id) ,
    unique key (email_address)
);
```

```
ALTER TABLE customers
CHANGE COLUMN number_of_complaints number_of_complaints INT DEFAULT 0;
```

```
ALTER TABLE customers
ALTER COLUMN number_of_complaints DROP DEFAULT;
```

Query

Write a SQL code to retrieve information desired from the database or insert, update, or delete data from it
Always put a semicolon at the end of the SQL Query

Operators

1. SQL Arithmetic Operators
2. SQL Comparison Operators
3. SQL Logical Operators

Operators

SQL Arithmetic Operators

Operators	Descriptions	Examples
+	It is used to add containing values of both operands	a+b will give 150
-	It subtracts right hand operand from left hand operand	a-b will give -50
*	It multiply both operand's values	a*b will give 5000
/	It divides left hand operand by right hand operand	b/a will give 2
%	It divides left hand operand by right hand operand and returns reminder	b%a will give 0

Operators

SQL Comparison Operators

Operator	Description	Example
=	Examine both operands value that are equal or not, if yes condition become true.	(a=b) is not true
!=	This is used to check the value of both operands equal or not, if not condition become true.	(a!=b) is true
< >	Examines the operand's value equal or not, if values are not equal condition is true	(a<>b) is true
>	Examine the left operand value is greater than right Operand, if yes condition becomes true	(a>b) is not true
<	Examines the left operand value is less than right Operand, if yes condition becomes true	(a
>=	Examines that the value of left operand is greater than or equal to the value of right operand or not, if yes condition become true	(a>=b) is not true
<=	Examines that the value of left operand is less than or equal to the value of right operand or not, if yes condition becomes true	(a<=b) is true
!<	Examines that the left operand value is not less than the right operand value	(a!
!>	Examines that the value of left operand is not greater than the value of right operand	(a!>b) is true

Operators

SQL Logical Operators

Operator	Description
ALL	this is used to compare a value to all values in another value set.
AND	this operator allows the existence of multiple conditions in an SQL statement.
ANY	this operator is used to compare the value in list according to the condition.
BETWEEN	this operator is used to search for values, that are within a set of values
IN	this operator is used to compare a value to that specified list value
NOT	the NOT operator reverse the meaning of any logical operator
OR	this operator is used to combine multiple conditions in SQL statements
EXISTS	the EXISTS operator is used to search for the presence of a row in a specified table
LIKE	this operator is used to compare a value to similar values using wildcard operator

Query

When querying data from a table when we have AND & OR condition in the WHERE clause then precedence is given to AND regardless of the order of using OR or AND

Precedence to AND & then OR
AND > OR

Wildcard Characters

% - Substitute for sequence of characters

_ - Substitute for single character

* - To get all columns & to count number of rows

Wildcard Characters

Symbol	Description	Example
%	Represents zero or more characters	bl% finds bl, black, blue, and blob
_	Represents a single character	h_t finds hot, hat, and hit
[]	Represents any single character within the brackets	h[oa]t finds hot and hat, but not hit
^	Represents any character not in the brackets	h[^oa]t finds hit, but not hot and hat
-	Represents a range of characters	c[a-b]t finds cat and cbt