SQL

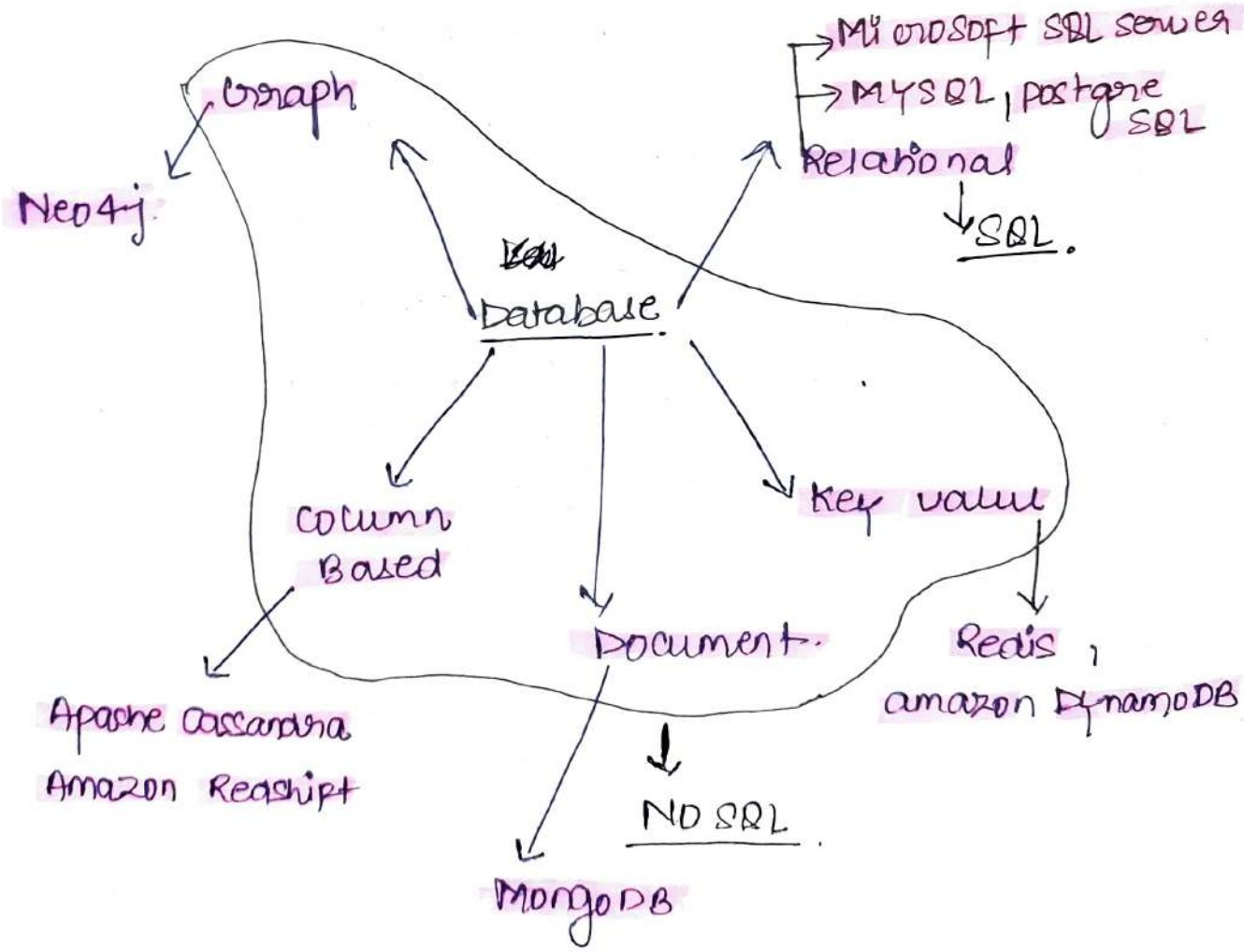
Data with Baraa

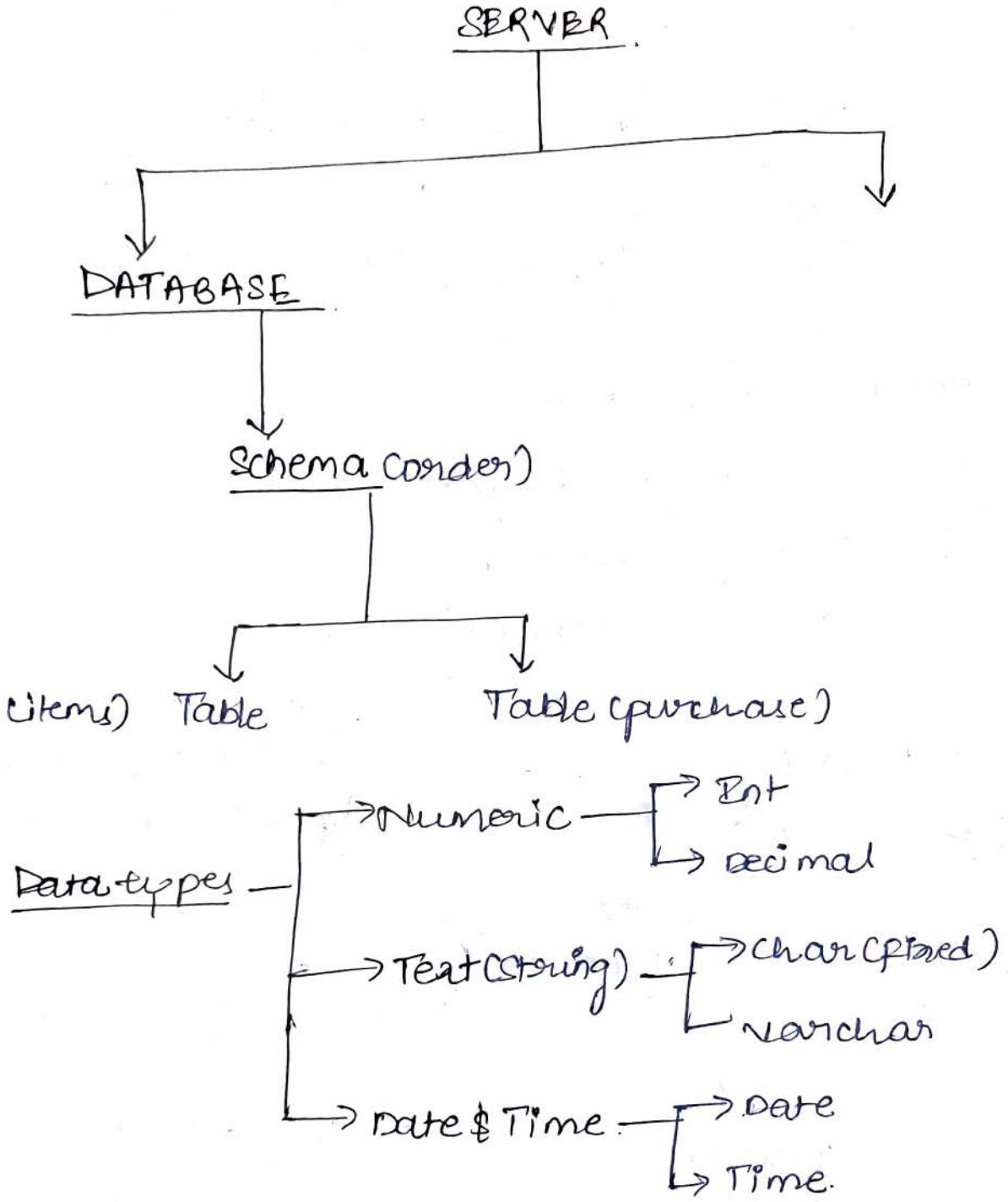
Beginner Level

Structured Query Language

- Secure
- Big Data handle
- Easy to Manipulate

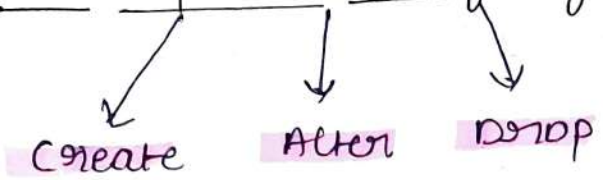
- Database stores Data
- SQL speak to Database
- DBMS Manages Database
- Server where database lives.



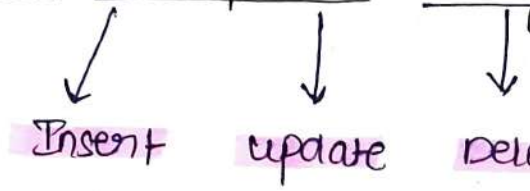

SERVER
DATABASE
Schema (order)
(items) Table
Table (purchase)
Data types
Numeric
Int
decimal
Text (String)
char (fixed)
varchar
Date & Time
Date
Time.

TYPES of SQL commands

- Data Depination Language (DDL)



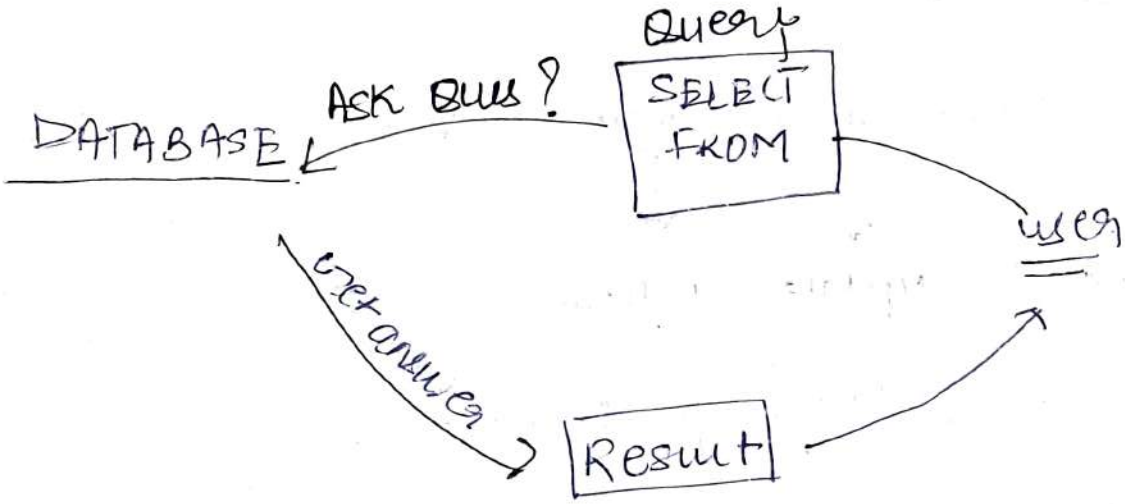
- Data Manipulation Language (DML)



- Data Query language :- SELECT

SQL SELECT

Ask your Data



SQL Query

SELECT *

- Choose Columns

  - SELECT * (ALL)

  Retrieves All Columns (Everything)

  - FROM

  Tells SQL where to find your Data.

For comment use.

-- comment

/* comment */

- SELECT FEW COLUMNS

pick only the columns you Needed instead to all.

② SELECT
    COL1,
    COL2,
① FROM Table.

Example.

```
SELECT
    name,
    country ──→ No comm after last column.
FROM customer.
```

- ~~SELECT~~ SQL QUERY WHERE

- Filter your Data.

where :- Filter your based on a condition.

③ SELECT *
① FROM Table
② WHERE condition.

Example. OR

③ SELECT
name,
country
① FROM Table
② where condition.

Example • Score not equal to 0.

SELECT *
FROM customers
WHERE Score != 0
condition.

• customer from Germany

SELECT *
FROM customer
WHERE country = 'Germany'
condition.

String should be in single quotes.

- SQL Query - order By

↓ sort your data.

③ SELECT *

① FROM Table.

② ORDER BY score DESC

↓

default order is ASC

Example. Sort the Result highest score first

```
SELECT *
    FROM customers
    ORDER BY score DESC
```

↓ Descending.

- Nested corder By)

```
SELECT *
    FROM customers
    ORDER BY country ASC, score
        DESC
```

• SQL Query – Group By

↓

– aggregate your data

combines rows with same values.

Aggregates a column by another column.

Total Score By country

Syntax.

③ SELECT category.
country,
SUM(Score) — aggregation.
① FROM Table
② Group By country

Eg:- find Total Score by each country.

SELECT
country,
SUM(Score) As Total-Score
FROM customer
Group By country.

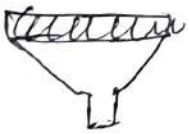
For column name.

Rule :- All columns in the SELECT must be either aggregated or included in GROUP BY.

- SQL Query - HAVING
  - Filter aggregate Data.
  - Filter Data after aggregation.
  - can be used only with GROUP BY.

Syntax

④ SELECT
country,
SUM(score)
① FROM Table
② GROUP BY country
③ Having condition
SUM(score) > 800



Filter your Data

① Before aggregation.

② After aggregation.

⑤ SELECT
country,
SUM(score)
① FROM Table
② WHERE score 7400
③ GROUP BY country
④ HAVING SUM(score) > 800

Before aggregation. ←

Ex. Find avg score per each country considering only customer with score not equal to 0 and return only those countries with avg(score) > 430.

```
SELECT
    Country,
    AVG(score) AS avg_score
FROM customers
WHERE score != 0
GROUP BY country
HAVING AVG(score) > 430.
```

- SQL Query – DISTINCT

↓

Remove duplicates

↓

Repeated values

(2) SELECT DISTINCT

country

(1) FROM Table

Example :- Return unique list of all countries.

```
SELECT DISTINCT
    country
FROM customers.
```

- Top – SQL Query

↓

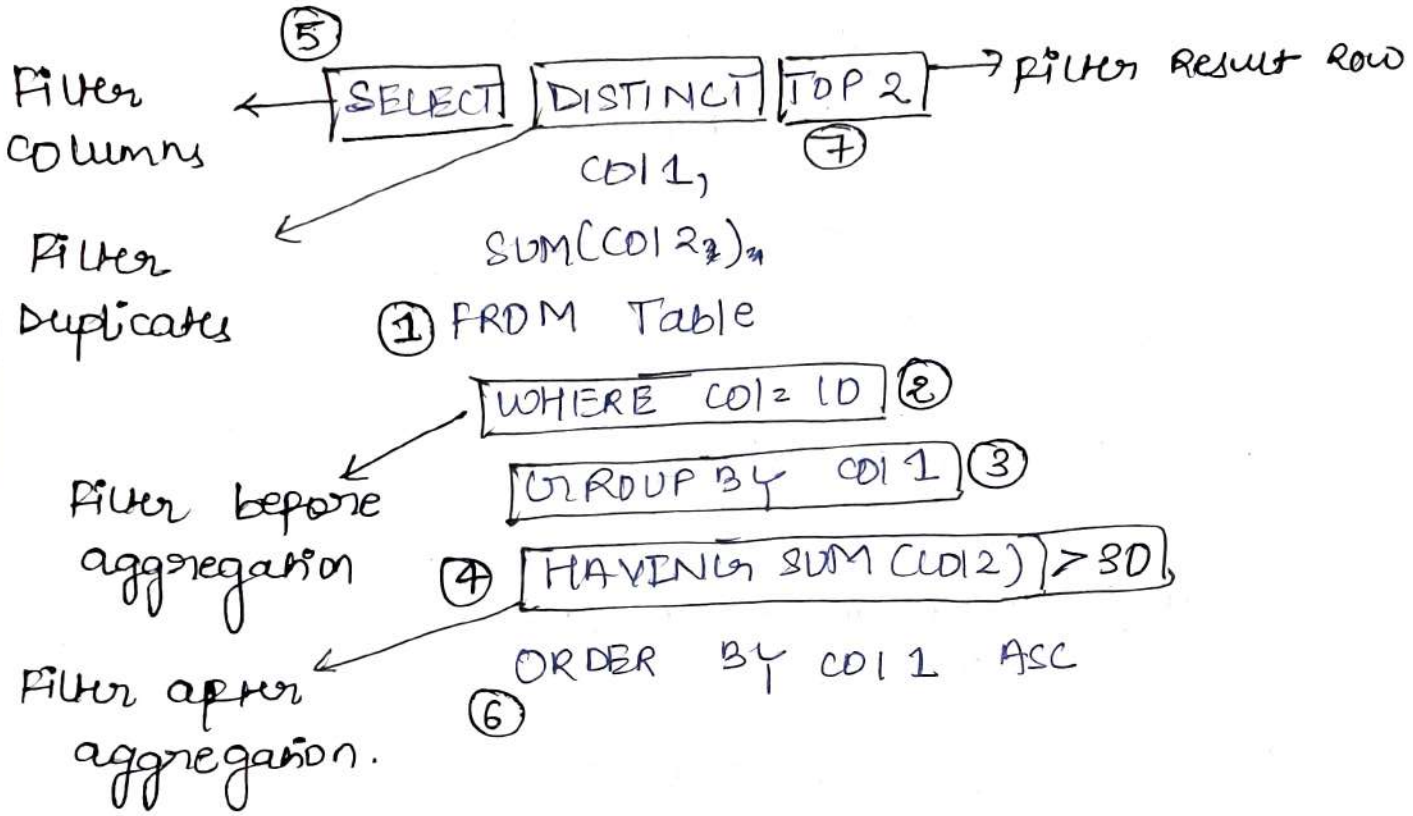
Restrict the number of items Returned.

(2) SELECT (3) TOP 3

*

(1) FROM Table

Execution order v/s coding order.



Execute order

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT DISTINCT
6. ORDER BY
7. TOP

Data definition language (DDL)

#CREATE

Create a new table called persons with columns: id, person_name, birth_date, and phone.

```
CREATE TABLE persons (
    id NOT NULL,
    person_name VARCHAR(50) NOT NULL,
    birth_date DATE,
    phone VARCHAR(15) NOT NULL,
    CONSTRAINT pk_person PRIMARY KEY (id)
)
```

ALTER.

Add new column called Email to the persons table.

```
ALTER TABLE persons
ADD email VARCHAR(50) NOT NULL
```

```
ALTER TABLE persons
DROP COLUMN phone.
```

DROP

Remove Table

↙

Delete the Table persons prom Database.

DROP TABLE persons;

↓

it will delet all Table with Record.

DML

↓

Data Manipulation Language

INSERT

1) INSERT : Manual Entry (values)

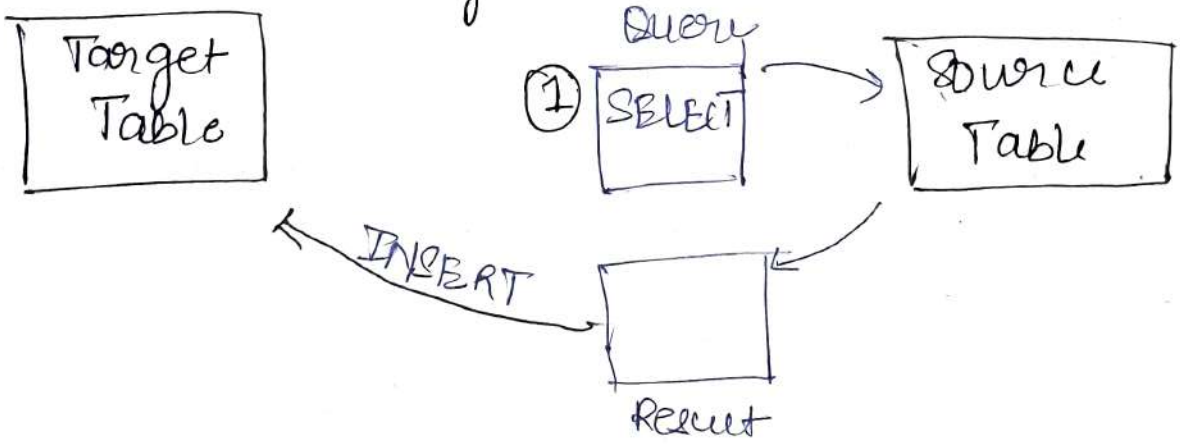
Syntax.

INSERT INTO table_name(col1, col2, col3, ---)
VALUES (value1, value 2, value 3)

Rule: Match the No. of colums and values.
we can do multiple Inserts.

- Constraints must be fulfilled.

2) Using - SELECT



INSERT USING SELECT

Ex:- Copy data from 'customers' table into 'persons'

insert data in persons table

```
INSERT INTO persons (id, person_name,
birth_date, phone)

SELECT
    id,
    First_name,
    NULL,
    'unknown'
FROM customers;
```

→ from customers Table.

#UPDATE

update syntax

```
UPDATE Table_name
SET col1 = value1,
    col2 = value2,
WHERE <condition>
```

[Note: Always use where to avoid updating all rows unintentionally.]

NULL :- unknown / no value

Ex:- change the customer score from 6 to 0.

```
UPDATE customers
SET score = 0 → do score 0
WHERE id = 6   where id is 6.
```

Ex:- update country and score

```
UPDATE customers
SET score = 0
    country = 'UK'
WHERE id = 10
```

we update two column where id is 10

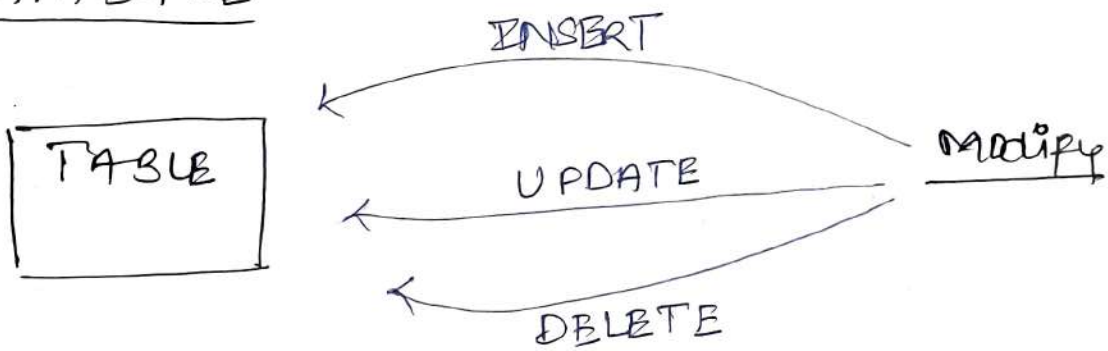
Ex:- update all customer with NULL score by setting their score to 0.

```
UPDATE customers
    SET score = 0
WHERE score IS NULL
```

→ score 0 karo jahan jahan value = NULL hai

DELETE

DATABASE



Delete syntax

```
DELETE FROM table_name
WHERE <CONDITION>
```

Note: Always use WHERE to avoid DELETING all rows unintentionally.

Ex:- Delete all customers with an ID greater than 5.

```
DELETE FROM customes   — Table name
WHERE id > 5
      ↓
   condition
```

Ex:- Delete all data FROM persons Table.

* TRUNCATE : Clears the whole table at once without checking or logging.

```
TRUNCATE TABLE persons.
```