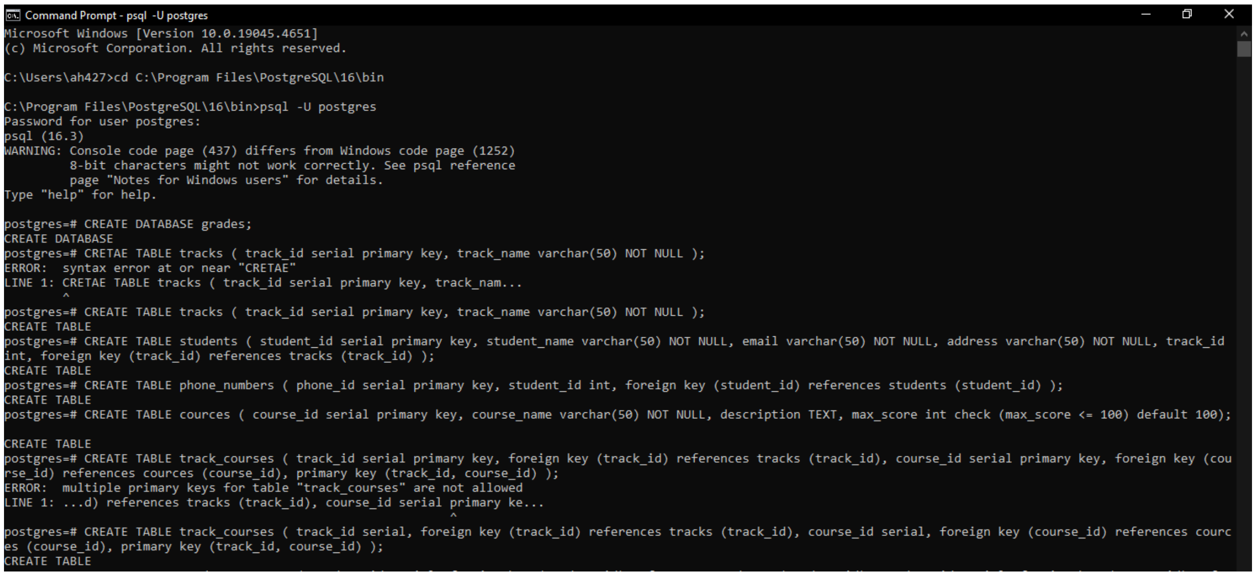
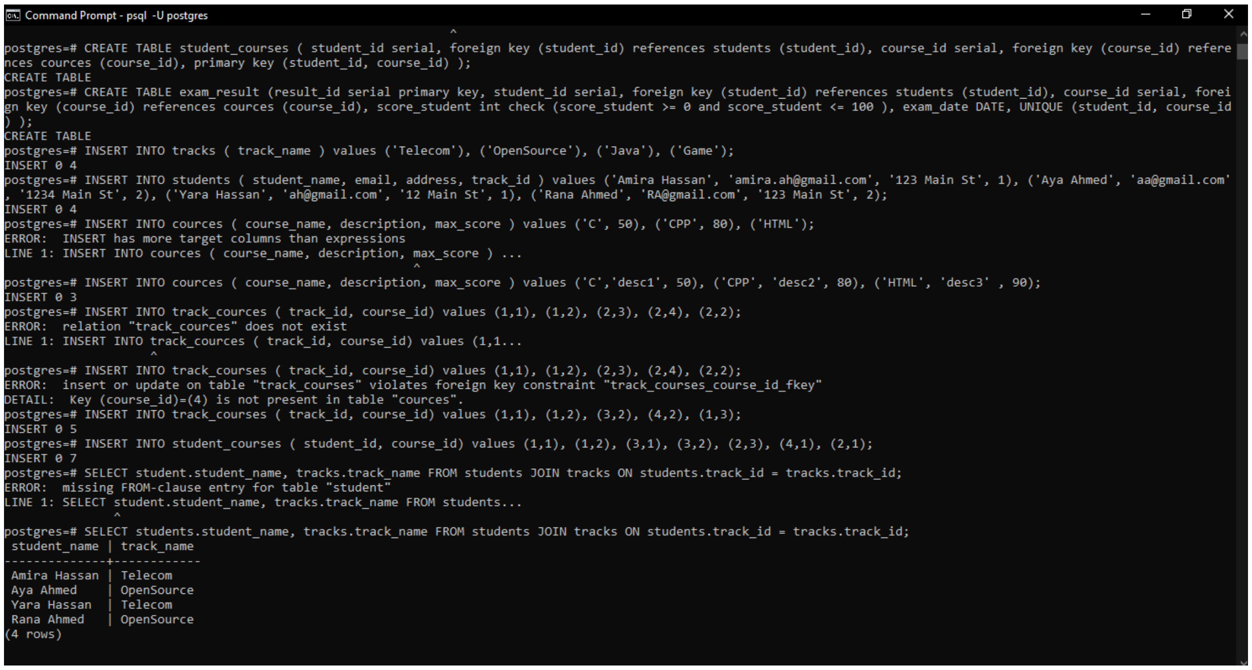
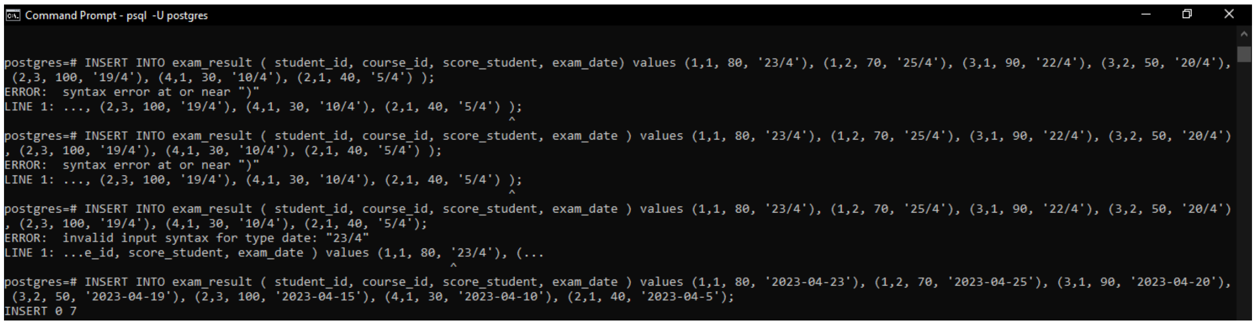
Commands



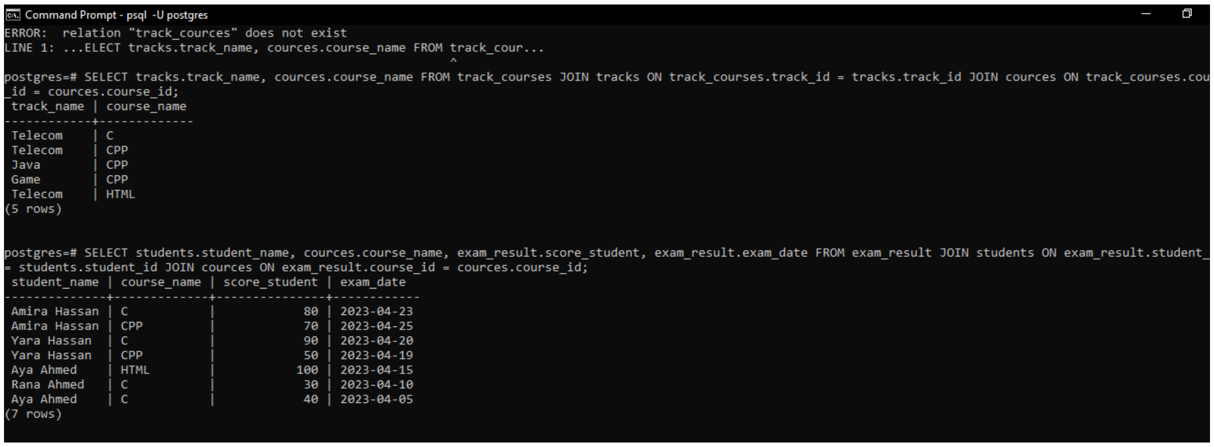
```
Command Prompt - psql  -U postgres
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ah427>cd C:\Program Files\PostgreSQL\16\bin

C:\Program Files\PostgreSQL\16\bin>psql -U postgres
Password for user postgres:
psql (16.3)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# CREATE DATABASE grades;
CREATE DATABASE
postgres=# CRETAE TABLE tracks ( track_id serial primary key, track_name varchar(50) NOT NULL );
ERROR:  syntax error at or near "CRETAE"
LINE 1: CRETAE TABLE tracks ( track_id serial primary key, track_nam...
        ^
postgres=# CREATE TABLE tracks ( track_id serial primary key, track_name varchar(50) NOT NULL );
CREATE TABLE
postgres=# CREATE TABLE students ( student_id serial primary key, student_name varchar(50) NOT NULL, email varchar(50) NOT NULL, address varchar(50) NOT NULL, track_id
int, foreign key (track_id) references tracks (track_id) );
CREATE TABLE
postgres=# CREATE TABLE phone_numbers ( phone_id serial primary key, student_id int, foreign key (student_id) references students (student_id) );
CREATE TABLE
postgres=# CREATE TABLE cources ( course_id serial primary key, course_name varchar(50) NOT NULL, description TEXT, max_score int check (max_score <= 100) default 100);

CREATE TABLE
postgres=# CREATE TABLE track_courses ( track_id serial primary key, foreign key (track_id) references tracks (track_id), course_id serial primary key, foreign key (cou
rse_id) references cources (course_id), primary key (track_id, course_id) );
ERROR:  multiple primary keys for table "track_courses" are not allowed
LINE 1: ...d) references tracks (track_id), course_id serial primary ke...
                                                             ^
postgres=# CREATE TABLE track_courses ( track_id serial, foreign key (track_id) references tracks (track_id), course_id serial, foreign key (course_id) references courc
es (course_id), primary key (track_id, course_id) );
CREATE TABLE
```



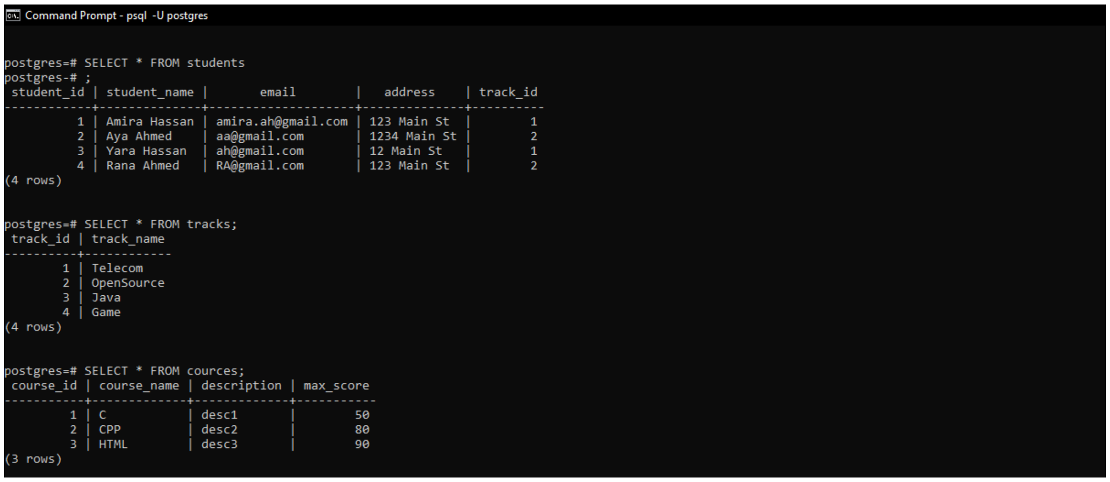
```
Command Prompt - psql  -U postgres
                                                             ^
postgres=# CREATE TABLE student_courses ( student_id serial, foreign key (student_id) references students (student_id), course_id serial, foreign key (course_id) refere
nces cources (course_id), primary key (student_id, course_id) );
CREATE TABLE
postgres=# CREATE TABLE exam_result (result_id serial primary key, student_id serial, foreign key (student_id) references students (student_id), course_id serial, forei
gn key (course_id) references cources (course_id), score_student int check (score_student >= 0 and score_student <= 100 ), exam_date DATE, UNIQUE (student_id, course_id
) );
CREATE TABLE
postgres=# INSERT INTO tracks ( track_name ) values ('Telecom'), ('OpenSource'), ('Java'), ('Game');
INSERT 0 4
postgres=# INSERT INTO students ( student_name, email, address, track_id ) values ('Amira Hassan', 'amira.ah@gmail.com', '123 Main St', 1), ('Aya Ahmed', 'aa@gmail.com'
, '1234 Main St', 2), ('Yara Hassan', 'ah@gmail.com', '12 Main St', 1), ('Rana Ahmed', 'RA@gmail.com', '123 Main St', 2);
INSERT 0 4
postgres=# INSERT INTO cources ( course_name, description, max_score ) values ('C', 50), ('CPP', 80), ('HTML');
ERROR:  INSERT has more target columns than expressions
LINE 1: INSERT INTO cources ( course_name, description, max_score ) ...
                                                        ^
postgres=# INSERT INTO cources ( course_name, description, max_score ) values ('C','desc1', 50), ('CPP', 'desc2', 80), ('HTML', 'desc3' , 90);
INSERT 0 3
postgres=# INSERT INTO track_cources ( track_id, course_id) values (1,1), (1,2), (2,3), (2,4), (2,2);
ERROR:  relation "track_cources" does not exist
LINE 1: INSERT INTO track_cources ( track_id, course_id) values (1,1...
                    ^
postgres=# INSERT INTO track_courses ( track_id, course_id) values (1,1), (1,2), (2,3), (2,4), (2,2);
ERROR:  insert or update on table "track_courses" violates foreign key constraint "track_courses_course_id_fkey"
DETAIL:  Key (course_id)=(4) is not present in table "cources".
postgres=# INSERT INTO track_courses ( track_id, course_id) values (1,1), (1,2), (3,2), (4,2), (1,3);
INSERT 0 5
postgres=# INSERT INTO student_courses ( student_id, course_id) values (1,1), (1,2), (3,1), (3,2), (2,3), (4,1), (2,1);
INSERT 0 7
postgres=# SELECT student.student_name, tracks.track_name FROM students JOIN tracks ON students.track_id = tracks.track_id;
ERROR:  missing FROM-clause entry for table "student"
LINE 1: SELECT student.student_name, tracks.track_name FROM students...
               ^
postgres=# SELECT students.student_name, tracks.track_name FROM students JOIN tracks ON students.track_id = tracks.track_id;
 student_name | track_name
--------------+------------
 Amira Hassan | Telecom
 Aya Ahmed    | OpenSource
 Yara Hassan  | Telecom
 Rana Ahmed   | OpenSource
(4 rows)
```



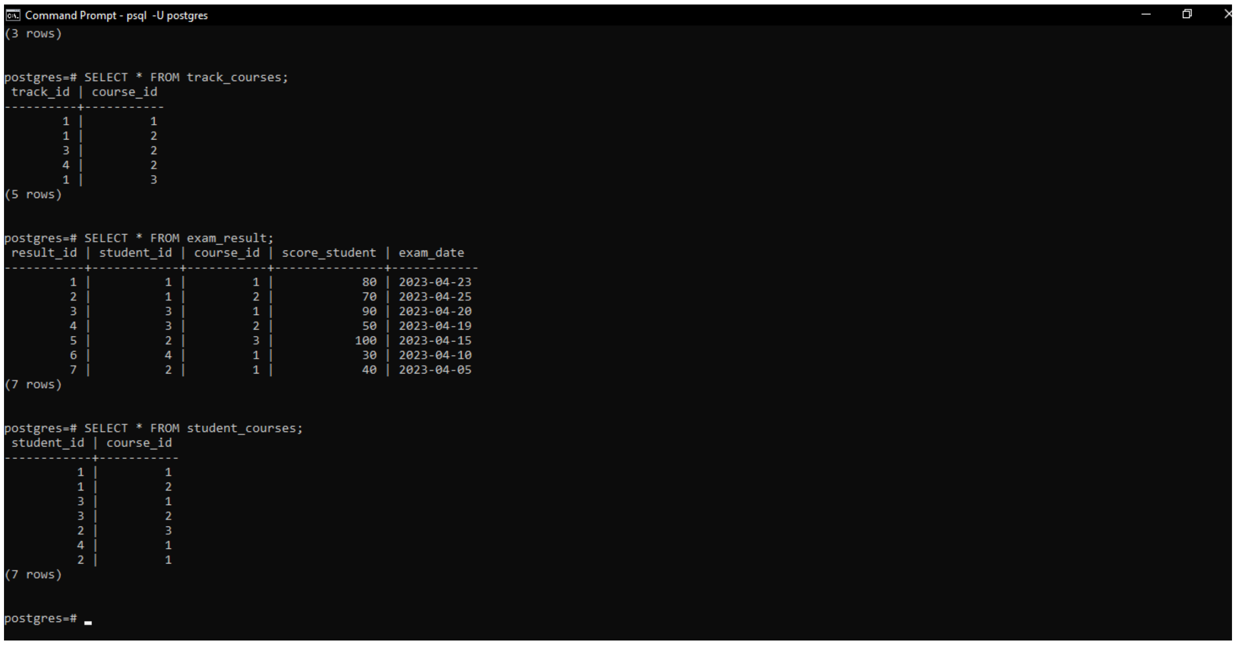
```
postgres=# INSERT INTO exam_result ( student_id, course_id, score_student, exam_date) values (1,1, 80, '23/4'), (1,2, 70, '25/4'), (3,1, 90, '22/4'), (3,2, 50, '20/4'),
 (2,3, 100, '19/4'), (4,1, 30, '10/4'), (2,1, 40, '5/4') );
ERROR:  syntax error at or near ")"
LINE 1: ..., (2,3, 100, '19/4'), (4,1, 30, '10/4'), (2,1, 40, '5/4') );
                                                                     ^

postgres=# INSERT INTO exam_result ( student_id, course_id, score_student, exam_date ) values (1,1, 80, '23/4'), (1,2, 70, '25/4'), (3,1, 90, '22/4'), (3,2, 50, '20/4')
, (2,3, 100, '19/4'), (4,1, 30, '10/4'), (2,1, 40, '5/4') );
ERROR:  syntax error at or near ")"
LINE 1: ..., (2,3, 100, '19/4'), (4,1, 30, '10/4'), (2,1, 40, '5/4') );
                                                                     ^

postgres=# INSERT INTO exam_result ( student_id, course_id, score_student, exam_date ) values (1,1, 80, '23/4'), (1,2, 70, '25/4'), (3,1, 90, '22/4'), (3,2, 50, '20/4')
, (2,3, 100, '19/4'), (4,1, 30, '10/4'), (2,1, 40, '5/4');
ERROR:  invalid input syntax for type date: "23/4"
LINE 1: ...e_id, score_student, exam_date ) values (1,1, 80, '23/4'), (...
                                                             ^

postgres=# INSERT INTO exam_result ( student_id, course_id, score_student, exam_date ) values (1,1, 80, '2023-04-23'), (1,2, 70, '2023-04-25'), (3,1, 90, '2023-04-20'),
 (3,2, 50, '2023-04-19'), (2,3, 100, '2023-04-15'), (4,1, 30, '2023-04-10'), (2,1, 40, '2023-04-5');
INSERT 0 7
```



```
ERROR:  relation "track_cources" does not exist
LINE 1: ...ELECT tracks.track_name, cources.course_name FROM track_cour...
                                                             ^

postgres=# SELECT tracks.track_name, cources.course_name FROM track_courses JOIN tracks ON track_courses.track_id = tracks.track_id JOIN cources ON track_courses.cou
_id = cources.course_id;
 track_name | course_name
------------+-------------
 Telecom    | C
 Telecom    | CPP
 Java       | CPP
 Game       | CPP
 Telecom    | HTML
(5 rows)


postgres=# SELECT students.student_name, cources.course_name, exam_result.score_student, exam_result.exam_date FROM exam_result JOIN students ON exam_result.student_
= students.student_id JOIN cources ON exam_result.course_id = cources.course_id;
 student_name | course_name | score_student | exam_date
--------------+-------------+---------------+------------
 Amira Hassan | C           |            80 | 2023-04-23
 Amira Hassan | CPP         |            70 | 2023-04-25
 Yara Hassan  | C           |            90 | 2023-04-20
 Yara Hassan  | CPP         |            50 | 2023-04-19
 Aya Ahmed    | HTML        |           100 | 2023-04-15
 Rana Ahmed   | C           |            30 | 2023-04-10
 Aya Ahmed    | C           |            40 | 2023-04-05
(7 rows)
```



```
postgres=# SELECT * FROM students
postgres-# ;
 student_id | student_name |       email        |   address    | track_id
------------+--------------+--------------------+--------------+----------
          1 | Amira Hassan | amira.ah@gmail.com | 123 Main St  |        1
          2 | Aya Ahmed    | aa@gmail.com       | 1234 Main St |        2
          3 | Yara Hassan  | ah@gmail.com       | 12 Main St   |        1
          4 | Rana Ahmed   | RA@gmail.com       | 123 Main St  |        2
(4 rows)


postgres=# SELECT * FROM tracks;
 track_id | track_name
----------+------------
        1 | Telecom
        2 | OpenSource
        3 | Java
        4 | Game
(4 rows)


postgres=# SELECT * FROM cources;
 course_id | course_name | description | max_score
-----------+-------------+-------------+-----------
         1 | C           | desc1       |        50
         2 | CPP         | desc2       |        80
         3 | HTML        | desc3       |        90
(3 rows)
```



```
(3 rows)


postgres=# SELECT * FROM track_courses;
 track_id | course_id
----------+-----------
        1 |         1
        1 |         2
        3 |         2
        4 |         2
        1 |         3
(5 rows)


postgres=# SELECT * FROM exam_result;
 result_id | student_id | course_id | score_student | exam_date
-----------+------------+-----------+---------------+------------
         1 |          1 |         1 |            80 | 2023-04-23
         2 |          1 |         2 |            70 | 2023-04-25
         3 |          3 |         1 |            90 | 2023-04-20
         4 |          3 |         2 |            50 | 2023-04-19
         5 |          2 |         3 |           100 | 2023-04-15
         6 |          4 |         1 |            30 | 2023-04-10
         7 |          2 |         1 |            40 | 2023-04-05
(7 rows)


postgres=# SELECT * FROM student_courses;
 student_id | course_id
------------+-----------
          1 |         1
          1 |         2
          3 |         1
          3 |         2
          2 |         3
          4 |         1
          2 |         1
(7 rows)


postgres=# _
```

Tables

```
CREATE TABLE tracks (
                    track_id SERIAL PRIMARY KEY,
                    track_name VARCHAR(50)
                    );

 CREATE TABLE courses (
          course_id SERIAL PRIMARY KEY,
           course_name VARCHAR(50),
           description TEXT,
           max_score INTEGER CHECK (max_score = 100)
```

```
            );

CREATE TABLE students (
            student_id SERIAL PRIMARY KEY,
             student_name VARCHAR(50),
             email VARCHAR(50),
             address VARCHAR(255),
             track_id INTEGER REFERENCES tracks(track_id)
            );

CREATE TABLE phone_numbers(
             phone_id SERIAL PRIMARY KEY,
            student_id INTEGER REFERENCES students(student_id),
             Phone_num  varchar(50),
            );

 CREATE TABLE track_courses (
             track_id INTEGER REFERENCES tracks(track_id),
             course_id INTEGER REFERENCES courses(course_id),
             PRIMARY KEY (track_id, course_id)
            );

CREATE TABLE student_courses (
             student_id INTEGER REFERENCES students(student_id),
             course_id INTEGER REFERENCES courses(course_id),
             PRIMARY KEY (student_id, course_id)
```

```
    );


CREATE TABLE exam_results (
    student_id INTEGER REFERENCES students(student_id),
     course_id INTEGER REFERENCES courses(course_id),
     score_student INTEGER CHECK (score_student >= 0 AND score_student <= 100),
     exam_date DATE, PRIMARY KEY (student_id, course_id)
    );
```

Reports

1. What is NoSQL?

- The term "NoSQL" stands for "Not Only SQL,"
- NoSQL database is non-relational database, designed to handle large volumes of data and use various data models such as document, key-value, column-family, and graph models.
- **Examples of NoSQL Databases**
    - **MongoDB**: A document-oriented database that stores data in JSON-like BSON documents.
    - **Cassandra**: A highly scalable column family store designed for high availability.
    - **Redis**: An in-memory key-value store known for its speed and used for caching and real-time analytics.
    - **Neo4j**: A graph database that excels in handling complex relationships and connected data.
    - **CouchDB**: A document store that uses JSON for documents, JavaScript for MapReduce queries, and HTTP for an API.

2. DBMSs Types?

- Hierarchical DBMS

    - Data is organized in a tree-like structure, where each record has a single parent.
    - Typically used in applications where the data relationships are well-defined, such as file systems, XML data, and organizational structures.
    - Efficient for one-to-many relationships.
    - Inflexible schema, difficult to manage relationships outside the hierarchy
    - **Example**: IBM's Information Management System (IMS).

- Network DBMS
    - Uses a graph structure where nodes (records) can have multiple parent and child relationships, allowing many-to-many relationships.
    - Suitable for applications with complex relationships.
    - More flexible than hierarchical DBMS.
    - Complex to design and manage so less common in modern systems.
    - Examples: Integrated Data Store (IDS), IDMS (Integrated Database Management System).

- Relational DBMS (RDBMS)
    - Data is organized in tables (relations) consisting of rows (tuples) and columns (attributes). Relationships between tables are established through foreign keys.
    - flexible querying with SQL, widely supported.
    - it can become complex with very large datasets, less efficient for unstructured data.
    - **Example**: MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server.

- Object-Oriented DBMS (OODBMS)
    - Data is stored as objects that contain both data and methods.
    - For applications requiring complex data representation.
    - Supports complex data types and relationships.
    - **Example**: ObjectDB, db4o.

- NewSQL DBMS

    - Combines the strengths of RDBMS and NoSQL, suitable for cloud environments.
    - **Example**: Google Spanner, CockroachDB, NuoDB.

- Distributed DBMS
    - Data is distributed across multiple locations or nodes, but appears as a single database to the user.
    - **Example**: Apache Cassandra, Google Spanner.