**­­­Session-1**

**DB Types**

1. Relational DB – if requirement is CRUD.
2. Non-Relational DB
   1. Document – if requirement is read data. (MongoDB)
   2. Wide Column
   3. Graph
   4. Key-Value – for caching (Redis, Memcached)

**SQL FEATURES**

1. Low Redundancy
2. Easy Maintenance of DB's
3. Enhanced Security
4. File Consistency
5. Multi-User Environment Support

**DATA MODELLING**

* It is abstraction of real-world system.
* Mapping Relations.
* Helps to visualize business.
* Reduce cost of change.
* Model is a means of communication.
* Model help elicit and document requirements.

**DATA MODEL TYPES**

* + **Conceptual Data Model** - process models, data reqs
    - It is a view of entities, attributes and their relations.
    - Ex: customer & product are 2 entities, cname & cnumber are attributes of customer entity. Sale is the relationship b/w customer & product.
  + **Logical Data Model** - technical reqs, performance reqs
    - Defines structure of data elements. (Attributes type with precision)
    - Primary/secondary key are not defined.
  + **Physical Data Model** - business data
    - Primary key, foreign key, views, indexes, access profiles, authorizations etc. defined.

**SQL Server**

* It also supports T-SQL (Transact-SQL), which is proprietary language of Microsoft.
* Provides extra features like declaring variable, exception handling, stored procedure etc.

**Files in SQL Server**

* + Primary File (.mdf) stores data about tables, views, triggers
  + Secondary File (.ndf) - optional and contain user-specific data
  + Log File (.ldf) - used for transaction management, using this we can perform rollback actions.

**Database**

* Collection of objects (tables, views, triggers, functions, stored procs etc.)

**SQL Command Categories**

* **DDL** create, alter, drop, truncate

|  |  |
| --- | --- |
| CREATE DATABASE <DB\_NAME> | Creates database |
| USE <DB\_NAME> | Uses db |
| SP\_HELPDB <DB\_NAME> | db description |
| SP\_HELP <TABLE\_NAME> | Table description |
| SP\_RENAME ‘OLD\_NAME’, ‘NEW\_NAME’ | Renames table |
|  |  |
| ALTER TABLE PRAC ADD UNIQUE(COL1,COL2) | Adds constraint |
| ALTER TABLE PRAC DROP CONSTRAINT <CONSTRAINT\_NAME> |  |
| ALTER TABLE PRAC ADD CONSTRAINT <CONSTRAINT\_NAME> CHECK(COND) |  |
|  |  |

* DML insert, update, delete
* TCL
* DCL grant, revoke

**Datatypes**

* bit, tinyint, smallint, int, bigint, decimal(p, s), numeric(p, s)
* date[yyyy-mm-dd], datetime2[yyyy-mm-dd hh:mm:ss], time[hh:mm:ss]
* char(n), varchar(n)

**KEYS IN DB**

* **Candidate Key :** attribute or set of attributes that can be uniquely identify a tuple
* **Super Key :** combination of two or more attributes identifies a tuple
* **Primary Key :** values should be unique and null not allowed
* **Alternate Key :** key other than primary key
* **Foreign Key :** column of one table points to the primary key of other table

**Commands**

SP\_HELPDB <DB\_NAME> (displays db info)

**Constraints**

* **NOT NULL :** null value can’t be accepted
* **UNIQUE :** values should be unique allows null

**CREATE TABLE TAB\_NAME(**

**CONSTRAINT CON\_NAME UNIQUE(COL1, COL2)**

**);**

* **CHECK :** checks value satisfies the condition or not

**CREATE TABLE TAB\_NAME(**

**COUNTRY VARCHAR(10) CHECK(COUNTRY = ‘INDIA’)**

**);**

* **DEFAULT :** default value if no value is specified

**CREATE TABLE TAB\_NAME(**

**COUNTRY VARCHAR(10) DEFAULT ‘INDIA’**

**);**

* **INDEX :** used to create index & can retrieve data from db quickly

**CREATE INDEX <INDEX\_NAME> ON TABLE\_NAME(COLUMNS\_NAME)**

**STORED PROCEDURES:**

CREATE PROCEDURE <PROC\_NAME>

AS

SQL\_STATEMENT

GO;

EXEC <PROC\_NAME>;

**NORMALIZATION**

* Process of reducing data redundancy
* **1NF**
  + Remove repeating groups from table
  + Create separate table for each set of related data
  + Identify primary key in all tables
* **2NF**
  + Table should not contain partial dependency.
    - i.e. if tables has more than 1 primary key, create multiple tables.
* **3NF**
  + There should be no transitive dependency for non-prime attributes
    - i.e,

**Spring Boot db connection**

spring.datasource.url=jdbc:sqlserver://localhost:1433;databaseName=sam;encrypt=true;trustServerCertificate=true;  
spring.datasource.username=sa  
spring.datasource.password=hello0006@  
spring.jpa.show-sql=true  
spring.jpa.properties.hibernate.format\_sql = true  
  
spring.jpa.hibernate.ddl-auto = update