Evolution, DDL, DML, DCL

Database Management System

Sujan Tamrakar

Evolution

- Early db application maintained in large organization had large number of records with similar structure.
  - University application: similar info. for each student, staff, course, grade records.
  - Hospital application: categorized info. about doctor, nurse, patient, staffs, equipment, lab, services.
- Issues:
  - Main problem with old types of db system was intermixing of conceptual relationships and improper placement of records on disk.
  - Problem in accessing records efficiently when new queries were identified.
  - New queries required different storage organization for efficient processing.
  - Problem in reorganizing the db when changes were made.
  - Only provision of PLI (Programming Language Interface)
  - New program had to be written, tested and debugged if new queries were to be implemented.
  - Mostly db system were used on large and expensive mainframe computers.

Evolution

- Thus, came existence of Relational DB.
- Relational db model introduced high level query languages that provided an alternative to PLI. Hence, it was a lot quicker to write new queries.
- Early RDBMS were slow because they did not use record placement/pointer to access related data.
- With the development of new storage and indexing techniques, better query processing, optimization and boost in performance were observed.
- Therefore, RDBMS eventually became dominant type of DBMS (nowadays as well).

Evolution

customerId	customerName	customerStreet	customerCity
C1015	Sam	Lamachaur	Pkr
C1016	Gopal	Batulechaur	Pkr
C1017	Muna	NewRoad	Ktm
C1018	Madan	NewRoad	Pkr

Customer table

Account table

accountNo	balance
A101	10000
A105	5000
A107	50000
A109	25000

Depositor table

customerId	accountNo
C1015	A101
C1017	A105
C1015	A107
C1018	a109

Fig: A sample relational database

DDL

- Data Definition Language
- Defines/modifies db **schema** or structure
- Specifies conceptual and internal schemas for db and any mappings between the two.
- DBMS will have DDL compiler which processes DDL statements in order to identify schema descriptions and store schema description in DBMS catalogue.
- It accepts data definition in sources form and converts them into object form.

DDL

- In most DBMS:
  - DDL defines user views, storage structures.
- In some DBMS:
  - Separate languages (VDL, SDL) may exist for specifying views and storage structure.
  - VDL = ‘View DL’ specifies user views and their mappings to conceptual schema
  - SDL = ‘Storage DL’ specifies internal schema.
- Ex:

Create: create objects Truncate: remove all records from table

Alter: change structure of db Rename: rename an object name

Drop: delete object from db Comment: add comment to data dictionary

DDL

Create: create objects

```
CREATE DATABASE my_database;


CREATE TABLE Students
    (
    ROLL_NO int(3),
    NAME varchar(20),
    SUBJECT varchar(20),
    );
```

DDL

Truncate: remove all records from table

```
TRUNCATE TABLE Student_details;
```

Drop: delete object from db

```
DROP TABLE Student_details;
DROP DATABASE student_data;
```

Rename: rename an object name

```
RENAME TABLE Countries TO Country;
```

DDL

Alter: change structure of db

```
ALTER TABLE Student RENAME COLUMN NAME TO FIRST_NAME;
ALTER TABLE Student RENAME TO Student_Details;
ALTER TABLE Student ADD (AGE number(3),COURSE varchar(40));
ALTER TABLE Student MODIFY COURSE varchar(20);
ALTER TABLE Student DROP COLUMN COURSE;
ALTER TABLE table_name ALTER COLUMN column_name column_type;
```

Comment: add comment to data dictionary

```
SELECT * FROM customer; -- Selects all columns and rows
SELECT * FROM customer; {Selects all columns and rows}
SELECT * FROM customer; /*Selects all columns and rows*/
```

DML

- Data Manipulation Language
- Used for managing ‘data’ within schema objects.
- Related with **instance**
- Used for specifying db retrievals and updates
- Uses CRUD (Create-Read-Update-Delete) operations and more.
- Handled by DML compiler
- 2 main types of DML;
  1. High level (non procedural)
     - Specifies complex db operations
     - Statements are either entered interactively or embedded in PL
     - Requires a user to specify WHAT data are needed without specifying how to get it.

DML

2. Low level (procedural)
   - Must be embedded in PL
   - Retrieves individual records from db & process each separately
   - Needs PL constructs such as looping
   - A.k.a. Record-at-a-time DML
   - Requires user to specify WHAT data are needed & HOW to get those data.

- Whenever DML commands (either high-level or low-level) are **embedded** in general purpose PL, that language is called Host language & such DML is called Data Sublanguage.
- High-level DML used in standalone interactive manner is called Query language.

DML

Ex:

1. SELECT
2. INSERT
3. DELETE
4. UPDATE
5. MERGE
6. CALL

DCL

- Data Control Language
- Control access to data and database using statements like:
  - GRANT= allow specified users to perform specified tasks.
  - REVOKE = cancel previously granted or denied permissions.
- The operations for which privileges may be granted to or revoked from a user or role apply to both the Data definition language (DDL) and the Data manipulation language (DML).

Ex:

- GRANT CREATE TABLE TO username;
- GRANT sysdba TO username;
- GRANT DROP ANY TABLE TO username;
- REVOKE CREATE TABLE FROM username;

Thank you