Chapter 1: Introduction to Database

Analytics Tensor
Mahesh KC
mahesh.kc@analyticstensor.com
https://analyticstensor.com

Introduction

- Database
- Terminology
- OLTP and OLAP
- Business use case for Financial Institution
- Key
- Relationships and Referential Integrity
- Normalization
- Denormalization
- Wrap-up

Database

A database is a container for objects that not only store data, but also enable data storage and retrieval to operate in a secure and safe manner. A database can hold the following objects:

- Table
- Columns
- Rows
- Stored procedures
- Triggers
- Indexes
- Functions
- Views

Terminology

Tables: Table stores data. Table contain information within rows and columns. Table is a mXn dimensional matrix. Table are created in database. There can be multiple database in a server. Table can be treated as entity or object in term of OOPs.

Various table can exit on database such a user table, system table and temporary table. User table are user defined or created table. System table store information about the databases and table properties. Temporary table stores data temporarily. The data from temporary will be deleted after the session ends.

Columns: Column provide a definition of each single pieces of information. Column is an attributes or properties of the entity/object. Table cannot be created without at least one column. Column need to be defined with data types while creating table in Relation Database compared to NoSQL Database.

Rows: Row is a single unit of information. Rows are the entries that defines the attributes. Rows are also called ***records***. There can be any number of rows in a table unless the disk space is not an issue.

Terminology (cont.)

Stored procedures: Stored procedures is a combination of SQL statements. Stored procedures are stored in the database. Stored proc is a programmatic way to get the answer/information from the data.

Indexes: Index consist of one or more columns from the table while creating. Index helps to query the data faster. Index are build to optimize the table for faster execution and query performance. Index are defined before or after table creation.

Functions: A function is similar to a stored procedure, but it takes information one row at a time or produces information one row at a time. Function can be either User Define or User Defined Table.

Views: View is a virtual tables. Views can contain information combined from several tables and can present a more user-friendly interface to the data. Views can also add a great deal of security to an application. Views can also be indexed to speed processing of data within.

Metadata*:* Information about information. Data of data*.*

OLTP and OLAP

Online Transactional Processing (OLTP): An OLTP system provides instant updates of data. OLTP systems require many considerations to ensure they're fast and reliable and can keep the data integrity intact. Normally, OLTP systems are normalized to third normal form.

Online Analytical Processing (OLAP): An OLAP system is designed with the premise that the data remains fairly static with infrequent updates. These updates could be every night, weekly, monthly, or any other time variant as long as updates aren't happening on a frequent basis, like in an OLTP system. OLAP do not follow any design standards or normalization techniques. OLAP is used for analytical purpose. It is easier to take data and to slice and dice it without having to bring in data from a normalized table. OLAP contains flat table rather than several normalize table which also has fewer relationship and indexes.

Business Use Cases

Requirement Gathering.
Physical Data Model
Logical Data Model
ER Diagram
Database Design
Transactional System
Stored Procs

Keys

Key: A key is a way of identifying a record in a database table. Keys are used to build relationships between tables because a key refers to a whole record. Keys helps to uniquely identify a record in a table. A key can be defined on a single column if it identify the record if not can be specified to multiple columns.

Primary Key: The column defined with primary key must contain only unique values. It cannot allow NULL values. A primary key are used to link data from one table to data from another.

Foreign/Referencing Key: A foreign key is any key on a child table where a column(s) can be directly matched with exactly the same number and information from the master table. The primary-foreign key link is created through a constraint. Foreign key does not have to map to a primary key on a master table.

Candidate/Alternate Key: Table that contains more than one column(s) can acts as a primary key. These columns all possess the uniqueness property of a primary key. Null values are not allowed. Candidate keys are the candidate for being the primary key. A candidate key that is not the primary key of a table is called an alternate key. Zero or more alternate keys can be defined for a specific table.

Unique Constraint: A unique constraint is not a primary key, but the column(s) defined within the constraint could be a primary key. Unique constraint can contain NULL values. NULL is treated as any other value within a column. There can be multiple unique constraints but only one primary key in a table.

Relationship and Referential Integrity

Relation in database is a logical link between two tables. It binds the tables together in such a way that changes in one table do not cause data in another table to become invalid. A relationship is used to enforce data integrity. Various rules need to be followed to maintain the integrity of data. Referential integrity can also be enforced through stored procedures or triggers. Referential Integrity helps to maintain the data relationship between multiple tables.

Types of relationship:

1. One-to-one
2. One-to-many
3. Many-to-may

One-to-One: When one record in one table matches to just one record in another table.

One-to-Many: When one master record is linked with zero, one, or more records in a child table. It is the most common relationship found in a database.

Many-to-Many: When there are zero, one, or indeed many records in the master table related to zero, one, or many records in a child table.

Normalization

Normalizing helps to:-

- Minimize data redundancy.
- Make database efficient.

Drawback of normalization:–

- Create too many relationships.
- Create small tables.
- Many join are needed to retrieve one piece of information from multiple tables.

Types of Normal Forms:

- First normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)

Denormalization

Denormalization is the complete opposite of normalization. It contains data redundancy within a table to reduce the number of table joins and potentially speed up data access. Denormalization can be found in production systems where the join to a table is slowing down queries, or perhaps where normalization is not required. The drawback of denormalizing is that there will be duplicate and unnecessary information

Wrap-up

- Important points.
- Q & A.

Assignment-1

Describe the following:

- Key
- Primary Key
- Foreign/Referencing Key
- Candidate/Alternate Key
- Unique Constraint
- Referential Integrity

Assignment-2

Describe database normalization with example. (Required: 1NF, 2NF, 3NF, BCNF; Optional: 4NF, 5NF)

Use the link below to the find resources.

- https://support.microsoft.com/en-us/help/283878/description-of-the-database-normalization-basics
- https://mariadb.com/kb/en/library/database-normalization/
- https://www.guru99.com/database-normalization.html
- https://hackr.io/blog/dbms-normalization
- https://www.studytonight.com/dbms/database-normalization.php

References

https://dev.mysql.com/doc/refman/8.0/en/