**OLTP:**

**-**Online Transaction Processing

-class of software applications capable of supporting transaction oriented programs.

-ability to maintain concurrency

-they are decentralized to avoid single points of failure

-usually designed for a large number of users who conduct short transactions.

-database queries are usually simple

**Examples of OLTP:**

-Online banking

-Purchasing a book online

-Booking an airline ticket

-Sending a text message

-Order entry

-Telemarketers entering telephone survey results

-Call center staff viewing and updating customers’ details

**Characteristics of OLTP:**

-usually very specific in the task that they perform

-usually involve a single record or small selections of records

-indexed access to data

-large number of users

-frequent queries and updates

-fast response times

**Availability:**

**-**needs to have extremely availability

-it often deals with mission-critical data, and have a large number of users

**ACID Compliance:**

**-**In order to maintain data integrity, OLTP databases need to be ACID Compliant.

-and this ensures that the transactions are accurate, consistent, isolated and durable

**OLAP:**

**-**Online Analytical Processing

-a class of software programs which are characterized by relatively low frequency of online transactions.

-a category of database processing that facilitates business intelligence

-queries are often too complex and involve bunch of aggregations

-the effectiveness measure relies highly on response time

-widely used for data mining or maintaining aggregated historical data, usually in multi-dimensional schemas

**Examples:**

**-**provides analysts, managers, and executives with the information they need to make effective decisions about an organization’s strategic directions

-provide valuable insights into how their business is performing, as well as how they can make improvements

**Characteristics:**

**-**optimized for querying and reporting

-OLAP tools enable users to analyze multidimensional data interactively from multiple perspectives

-it can be used to find trends and get a big picture view of the data.

-it can also be used for complex number crunching, and to create “what if” scenarios for forward planning.

-OLAP applications include business reporting for sales, marketing, management reporting, business process management, budgeting and forecasting, financial reporting, and more.

-OLAP applications usually have smaller number of users.

**Example:**

OLTP application might be used to provide internet banking for a bank’s 1 million customers. But an OLAP app that analyzes those customer’s internet banking habits would be used by select number of bank staffs.

**OLAP Tools:**

-Dundas BI -IBM Cognos Analytics -SAP Business Intelligence

-Sisense -InetSoft -Halo

**OLTP vs OLAP:**

**-**OLAP has smaller number of user than OLTP

-OLAP involves large amounts of data, queries can be very large and run for much longer

-OLTP is concerned with speed, precision and supporting large number of users

-OLTP is concerned with inserting, updating, and deleting small amounts of data

-OLAP is concerned with complex aggregate queries across data sets

-OLAP databases are populated with batch queries, all data are inserted at the same time.

-OLTP is where data is constantly being inserted/updated/deleted

**OLAP and Data Warehouses:**

**-**They are two different things

-OLAP can be used to transform the data from a data warehouse into strategic information

-A data warehouse stores and manages data, typically in relational databases. They can be extremely large databases with enormous amounts of data

-OLAP is stored in a multidimensional database, where each data attribute is considered a separate “dimension”.

-OLAP tools can be used to extract data from the intersections of such dimensions.

-Example: all products sold in the Northeast region during the last quarter.

**OLAP Cubes:**

-It is a multidimensional array of data-sets

-products along x-axis, regions along y-axis and time period along z-axis

-OLAP tools could be used to summarize sale by product, regions or time period.

-Not restricted to three dimensions

-Can have any number of dimensions and they are called *hypercube.*

-Not actual cubes, because they might not have same length mathematically

**Data Warehouse:**

-It is a large collection of data that can be used to help an organization make key business decisions.

-A data warehouse is subject-oriented, integrated, nonvolatile and time-variant collection of data in support of management’s decisions

**Collation:**

-It refers to set of rules that determines how data is sorted and compared.

-Rules defining the correct character sequence are used to sort the character data.

**Types of Collation:**

-Case sensitivity: **A** and **a** are treated differently

-Accent sensitivity: Accent letter are treated differently

-Kana sensitivity: Hiragana vs Katakana

-Width sensitivity: Same character represented in single-byte (half width) vs double byte

**Stored Procedure**

-It is a subroutine available to applications that access RDBMS.

-They are stored in database data dictionary

-Disadvantage: it can be executed nowhere except in database and occupies more memory.

-Provides security and functionality

DELIMITER $$

CREATE PROCEDURE FetchAllStudents ()

BEGIN

SELECT \* FROM myDB.students;

END $$

DELIMITER;

**Recursive Stored Procedure:**

-A stored procedure which calls itself until a boundary condition is reached.

-helps to deploy same code several times when required

-Some SQL programming languages limit the recursion depth to prevent an infinite loop of procedure calls from causing a stack overflow, which slows down the system and may lead to system crashes.

DELIMITER $$ \*\*Set a new delimiter => $$\*\*

CREATE PROCEDURE calctotal ( \*\*Create Procedure \*\*

IN number INT, \*\* Input and Output variables \*\*

OUT number INT

) BEGIN

DECLARE score INT DEFAULT NULL; \*\* Set default value => score \*\*

SELECT awards FROM achievements

WHERE id = number INTO score;

IF score IS NULL THEN SET total = 0; \*\* Termination condition \*\*

ELSE

CALL calctotal (number + 1); \*\*Recursive Call \*\*

SET total = total + score;

END IF;

END $$ \*\* End of procedure \*\*

DELIMITER; \*\* Reset the delimiter \*\*

**How to create empty tables with the same structure as another table?**

SELECT \* INTO Students\_copy

FROM students WHERE 1=2

**Pattern matching:**

**-**Using % wildcard to perform a simple search

SELECT \* FROM students

WHERE first\_name LIKE ‘K%’

-Omitting the patterns using NOT keyword

SELECT \* FROM students

WHERE first\_name NOT LIKE ‘K%’

-Matching a pattern anywhere using % twice

SELECT \* FROM students

WHERE first\_name LIKE ‘%Q%’

-Using the \_ wildcard to match pattern at a specific position

SELECT \* FROM students

WHERE first\_name LIKE ‘\_\_K%’ \*\*K in third position\*\*

-Matching patterns for specific lengths

SELECT \* FROM students

WHERE first\_name LIKE ‘\_\_\_%’ \*\*has 3 or more letters \*\*

SELECT \* FROM students

WHERE first\_name LIKE ‘\_\_\_\_’ \*\*has exactly 4 letters \*\*