**CS 31 Database Management Programming Lecture 2**

**Online Transaction Processing (OLTP) System** – an operational database system available for, and dedicated to, transaction processing

Components of an OLTP system include the database, DBMS, and applications used to enter transactions. These might include point-of-sale systems, internet applications, and mail orders.

**Data Warehouse** – a store of enterprise data that is designed to facilitate management decision making

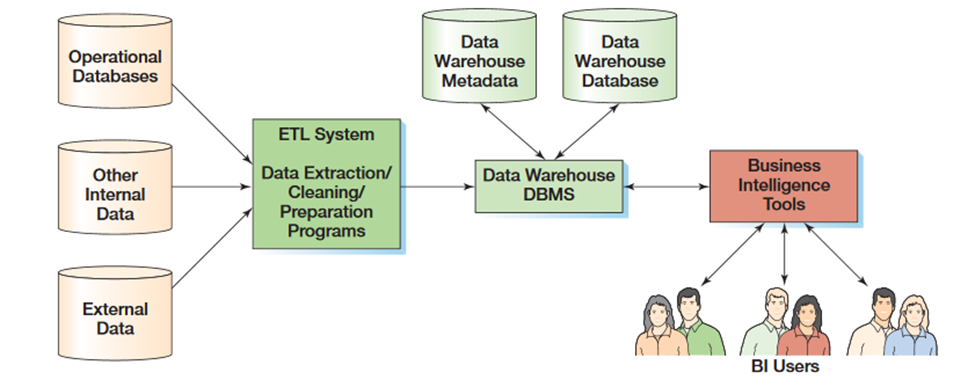
Components of a data warehouse include the database and DBMS. This is a different system than the DBMS used for the OLTP.

**Extract, Transform, and Load (ETL) System** – the portion of a data warehouse that converts operational data to data warehouse data

**Business Intelligence (BI) Systems** – information systems that assist managers and other professionals in the analysis of business activities and in the prediction of future events

Two major categories of BI systems are reporting systems and data mining systems. These systems will be linked to the data warehouse.

During the ETL process data may be split (date), columns omitted, columns aggregated, or prepared in other ways that are meaningful to the business unit.



The Cape Codd database is a data warehouse prepared for analysis by the marketing department.

**Cape Codd Tables**

* **RETAIL\_ORDER** table contains rows for each retail sales order
* **ORDER\_ITEM** table contains rows for each item in an order
* **SKU\_DATA** table contains rows for each stock-keeping unit (SKU)
* **WAREHOUSE** table contains rows for each warehouse
* **INVENTORY** table contains rows for each SKU present in a warehouse
* **CATALOG\_SKU\_20##** table contains rows for each SKU in the catalog and on the website for that year

You will sometimes see the following notation:

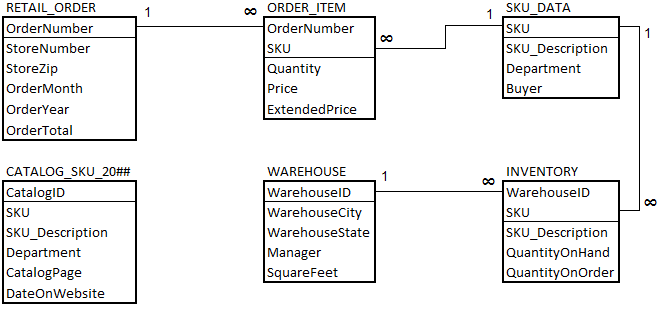
TABLE\_NAME(PrimaryKey, *ForeignKey*, ColumnData)

* Underlined – designates primary key, multiple underlined column names means composite primary key
* Italicized – designates foreign key, multiple underlined column names means composite foreign key
* Underlined and Italicized – primary and foreign key

**Schema** – complete logical view of the database, containing all the tables, all the columns in each table, the primary key of each table, and the foreign keys that link the tables together

**Cape Codd Schema**

* RETAIL\_ORDER(Order Number, StoreNumber, StoreZIP, OrderMonth, OrderYear, OrderTotal)
* ORDER\_ITEM(*OrderNumber*, *SKU*, Quantity, Price, ExtendedPrice)
* SKU\_DATA(SKU, SKU\_Description, Department, Buyer)
* WAREHOUSE(WarehouseID, WarehouseCity, WarehouseState, Manager, SquareFeet)
* INVENTORY(*WarehouseID*, *SKU*, SKU\_Description, QuantityOnHand, QuantityOnOrder)
* CATALOG\_SKU\_20##(CatalogID, SKU, SKU\_Description, Department, CatalogPage, DateOnWebsite)



**Structured Query Language (SQL)** – a data sublanguage developed by IBM in the 1970's. It's the universal query language of relational DBMS products, endorsed by ANSI (national standards) and ISO (international standards)

Five Categories of SQL Statements are:

* Data definition language (DDL) – statements for creating tables, relationships and other structures
* Data manipulation language (DML) – statements for querying, inserting, modifying, and deleting data
* SQL/Persistent Stored Modules (SQL/PSM) – statements, which extend SQL by adding procedural programming capabilities (variables, flow control statements)
* Transaction control language (TCL) – statements that are used to make transaction boundaries and control transaction behavior
* Data control language (DCL) – statements that are used to grant/revoke database permissions

**Graphical User Interface (GUI)** – an interface that uses graphical elements for interaction with a user

**Ad-hoc Query** – a query created by a user in SQL and used when needed, as compared to a predefined or stored query

We will be using MySQL Workbench (GUI) and MariaDB (DBMS) to perform ad-hoc queries.

**SQL/MySQL Workbench To Do List:**

1. Open Workbench and log in – password: root99
2. Run the Create and Insert Scripts from Canvas in Workbench
3. SHOW DATABASES; -- displays current databases

<https://dev.mysql.com/doc/workbench/en/wb-keys.html>

1. USE cape\_codd; -- selects the cape\_codd DB

Double-click name in Schemas pane

Right-click and "Set as default schema"

1. SHOW TABLES; -- shows tables of selected database
2. DESCRIBE table\_name; -- Shows properties of table\_name

SHOW COLUMNS FROM table\_name; -- Shows properties of table\_name

1. HELP keyword; -- Displays list of allowed keyword statements - works in command line only
2. Look at the  icons to the right of the table name in the Schemas pane.
3. Add a row the easy way
4. Delete a row the easy way
5. Rules about identifiers (database name, table name, and names of other structures you create) – treat them as case sensitive – depends on OS – check Workbench preferences: Edit -> Preferences -> General Editors -> Check "SQL Identifiers are Case Sensitive"
6. Include semicolons at the end of statements
7. Rules about SQL keywords – not case sensitive anywhere – Make auto-complete change them to all uppercase on completion: Edit -> Preferences -> SQL Editor -> Query Editor -> Check "Use UPPERCASE keywords on completion"
8. Show how to create new query tab
9. SQL script files have the .sql extension
10. Different ways to comment code in MySQL:

/\* multiline comments \*/

# single line comments

-- single line comments (space after second dash required)

1. Simple math calculations using SQL:

SELECT SQRT(16);

SELECT ABS(-1564);

SELECT PI();

SELECT 373 \* 89;

SELECT PI() \* SQRT(16);

1. The following are required SQL clauses when retrieving table data.

**SELECT** – specifies which columns are to be listed in the query results

**FROM** – specifies which tables are to be listed in the query results

1. The dot is used to further qualify a name.

**cape\_codd.sku\_data** is the fully qualified name for the sku\_data table.

**cape\_codd.sku\_data.SKU\_Description** is the fully qualified name for the SKU\_Description column of the sku\_data table.

1. **DISTINCT** works on all the columns listed
2. **LIMIT x, y – limit results to y rows starting at x**

**LIMIT y OFFSET x - limit results to y rows starting at x**

Row index starts at 0