# Database Training with Microsoft SQL Server (DAY – 1)

SummitWorks™
GLOBAL SOLUTION ARCHITECTS

Authorized & published by Summitworks Technologies Inc

# Agenda Day 1

- Software
  - Types of Software Applications
- Database
  - Types of database
  - Why to use Database
  - Logical relational data model and referential integrity concept
  - One to one, one to many, many to many
  - Data Types
  - Keys and Constraints
  - Normalization
  - Fundamental building blocks:
    - Table, columns, constraints
  - SQL server various versions
  - Installation of SQL Server Express on your computer/laptop
  - Transactional database versus analytical database

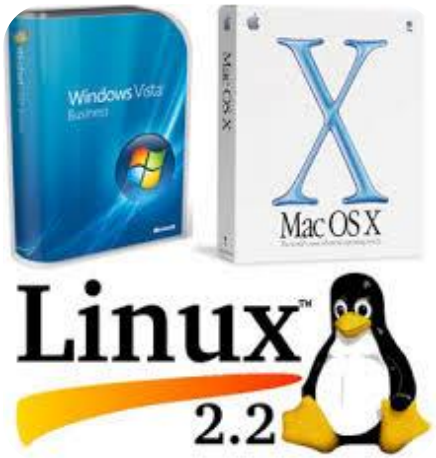
# Software

## What is a software?

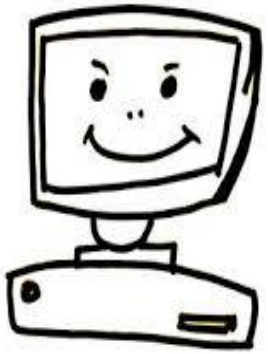
software

Application software


System software




hardware



- If hardware is the body, software is the mind
- In simpler terms, software provides steps by steps instructions for the hardware to operate.

# Different type of software applications

## Web based software:

- Software that requires Internet connection to run.

Examples:

- Google Maps
- Bing Maps
- Search Engines
- Social Media Applications

## Windows Based

- Software that runs native in your machine
  - Text editors such as MS-Word
  - Anti Virus
  - WinZip

## Apps

- Designs for Smart Devices such as Tablets or Phones
- IOS/Android & Windows App

## Business to Business (B2B), Business to Customer(B2C)

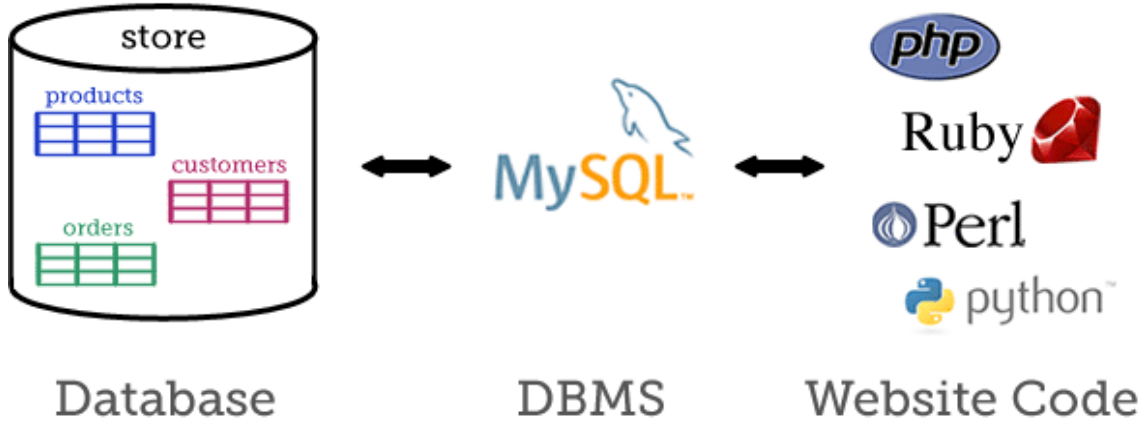
- Web Services/XML/EDI/X12 Standards
- Examples such as payment processing services

# Database

What is Database?

A *database* is a collection of information that is organized so that it can easily be accessed, managed, and updated.

- You can query data in a database in a very efficient way
- You can relate data from different tables using JOINs.
- Your data stored using a built-in structure.
- Databases are ACID(Atomicity, Consistency, Isolation, Durability)



# Benefits of Database

Why to use a database?

- Reduce the amount of time you spend managing data
- Analyze data in a variety of ways
- Standards can be enforced
- Security restrictions can be applied
- Improve the quality and consistency of information
- Eliminate or reduce Data Redundancy
- Data Integrity is maintained

# Types of Database

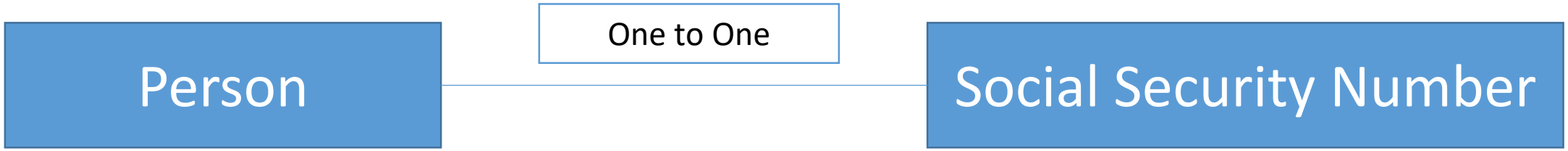
Different types of databases:

- Flat file
  - file containing records that have no structured interrelationship
  - Text files, Comma/Tab Delimited Files
- XML
  - E**X**tensible **M**arkup **L**anguage
  - Designed to be both human- and machine-readable
- Excel/Spreadsheets
  - Stores data in a tabular form with relationship
- Relational database
  - A database structured to recognize relations among stored items of information.

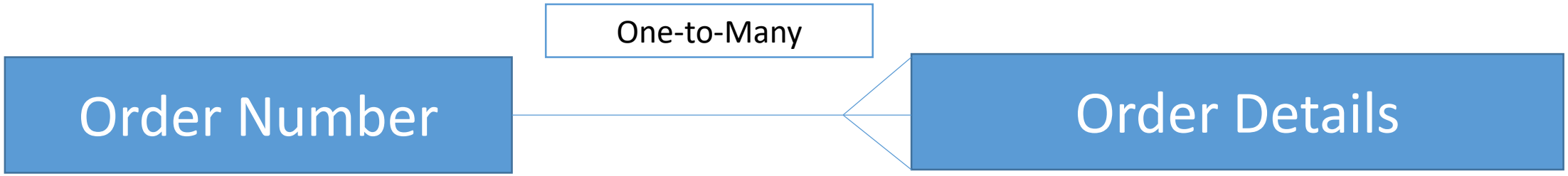
# Database Relationship

Logical relationship

- **One to One**
  - Both tables can have only one record on either side of the relationship
  - Each primary key value relates to only one (or no) record in the related table

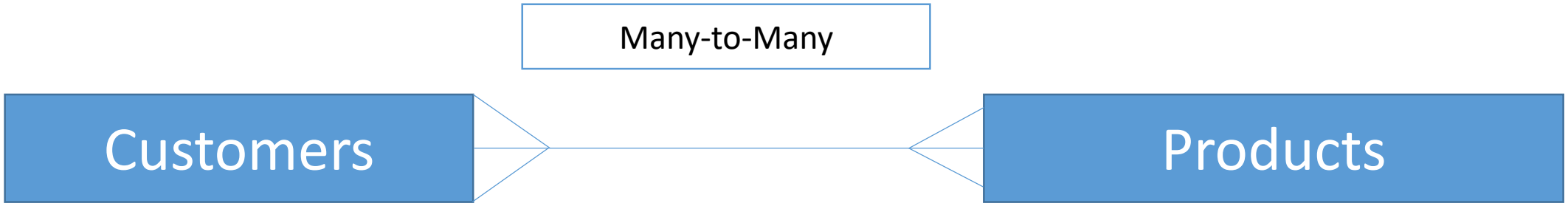



- **One-to-many**
  - The primary key table contains only one record that relates to none, one, or many records in the related table.



# Database Relationship

- **Many to Many**
  - Each record in both tables can relate to any number of records (or no records) in the other table.



# Data Types

Properly defining the fields in a table is important to the overall optimization of your database. You use only the type and size of field you really need to use.

MySQL uses many different data types broken into three categories −

- **Numeric:** INT , FLOAT
- **Date and Time:** DATE , DATETIME
- **String Types:** CHAR, VARCHAR() , VARCHAR2, BLOB or TEXT

# Data Types (Cont.)

**Numeric**

- **INT** – A normal-sized integer that can be signed or unsigned. If signed, the allowable range is from -2147483648 to 2147483647. If unsigned, the allowable range is from 0 to 4294967295. You can specify a width of up to 11 digits.
- **FLOAT(M,D)** – A floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 10,2, where 2 is the number of decimals and 10 is the total number of digits (including decimals). Decimal precision can go to 24 places for a FLOAT.
- **DOUBLE(M,D)** – A double precision floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 16,4, where 4 is the number of decimals. Decimal precision can go to 53 places for a DOUBLE. REAL is a synonym for DOUBLE.
- **DECIMAL(M,D)** – An unpacked floating-point number that cannot be unsigned. In the unpacked decimals, each decimal corresponds to one byte. Defining the display length (M) and the number of decimals (D) is required. NUMERIC is a synonym for DECIMAL.

# Data Types (Cont.)

**Date and Time (Temporal)**

- **DATE** – A date in YYYY-MM-DD format, between 1000-01-01 and 9999-12-31. For example, December 30th, 1973 would be stored as 1973-12-30.

- **DATETIME** – A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59. For example, 3:30 in the afternoon on December 30th, 1973 would be stored as 1973-12-30 15:30:00.

# Data Types (Cont.)

**String types**

- **CHAR(M)** – A fixed-length string between 1 and 255 characters in length (for example CHAR(5)), right-padded with spaces to the specified length when stored. Defining a length is not required, but the default is 1.

- **VARCHAR(M)** – A variable-length string between 1 and 255 characters in length. For example, VARCHAR(25). You must define a length when creating a VARCHAR field.

# Keys

**Keys** are important part of the arrangement of a table. Keys make sure to uniquely identify a table's each part or record of a field or combination of fields.

- Type of keys:
  - Natural Key
  - Surrogate/Artificial key
  - Candidate keys
  - Primary key
  - Secondary or Alternative key
  - Composite or Compound Key

# Keys (Cont.)

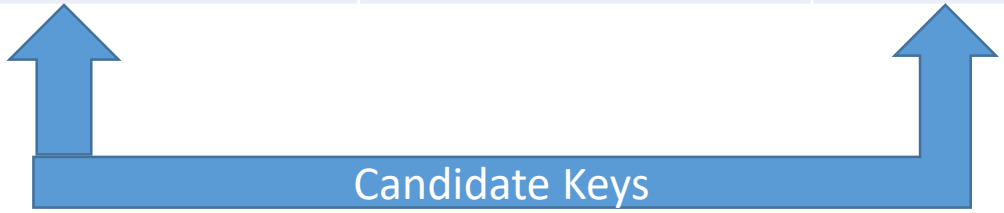
- **Natural Key:** It is a key that is naturally declared as the Primary key. Natural keys are sometimes called as business or domain keys because these key are based on the real world observation. So it is a key whose attributes or values exists in the real world. These attributes have logical relationship with the table.
  - For Example: Social Security Number (SSN) is a natural key that can be declared as the primary key.
- **Surrogate/Artificial key :** Surrogate key is artificially generated key and its main purpose it to be the primary key of table. Artificial keys do not have meaning to the table. There are few properties of surrogate or artificial keys.
  - They are unique because these just created when you don't have any natural primary key.
  - They are integer values. One cannot find the meaning of surrogate keys in the table.
  - End users cannot surrogate key. Surrogate keys are allowed when no property has the parameter of primary key.
  - The primary key is huge and complex.
  - Example: Table which has the details of the student has primary key but it is large and complex. The addition of row id column to it is the DBA's decision, where the primary key is row id.

# Keys (Cont.)

- **Candidate keys:** A Candidate Key is a set of one or more fields/columns that can identify a record uniquely in a table. There can be multiple Candidate Keys in one table. Each Candidate Key can work as Primary Key.

Example: In the below diagram Student Id, First Name and Last Name are Candidate Keys since all these three fields can be work as Primary Key.

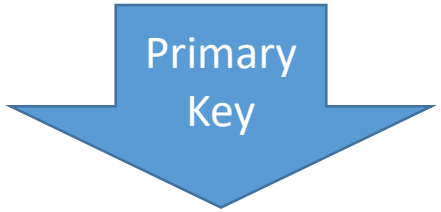
| Student Id | First Name | Last Name | Course Id |
|---|---|---|---|
| 1123 | Rose | Mary | 001 |
| 1124 | Jessy | Samuel | 002 |
| 1125 | John | Williams | 003 |
| 1126 | Lily | Holmes | 004 |



# Keys (Cont.)

- Primary Key: Primary key is a set of one or more fields/columns of a table that uniquely identify a record in a database table. It can not accept null, duplicate values. Only one Candidate Key can be Primary Key.

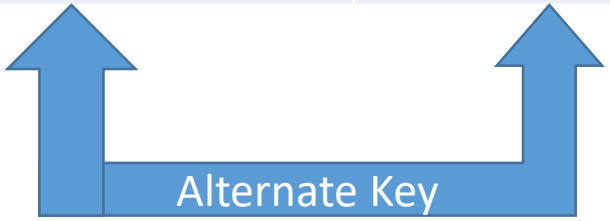
| Student Id | First Name | Last Name | Course Id |
|---|---|---|---|
| 1123 | Rose | Mary | 001 |
| 1124 | Jessy | Samuel | 002 |
| 1125 | John | Williams | 003 |
| 1126 | Lily | Holmes | 004 |



# Keys (Cont.)

- **Secondary or Alternate Key**: An Alternate key is a key that can be work as a primary key. Basically, it is a candidate key that currently is not a primary key.
- Example: In the below diagram First Name and Last Name become Alternate Keys when we define Student Id as Primary Key.

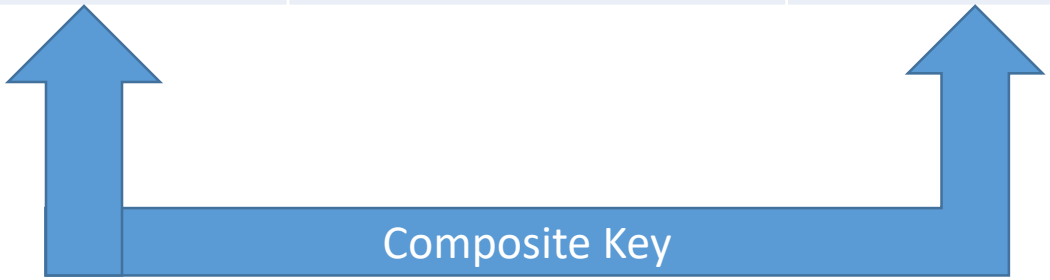
| Student Id | First Name | Last Name | Course Id |
|---|---|---|---|
| 1123 | Rose | Mary | 001 |
| 1124 | Jessy | Samuel | 002 |
| 1125 | John | Williams | 003 |
| 1126 | Lily | Holmes | 004 |



# Keys (Cont.)

- Composite or Compound Key: Composite Key is a combination of more than one fields/columns of a table. It can be a Candidate key, Primary key.

| Student Id | First Name | Last Name | Course Id |
|---|---|---|---|
| 1123 | Rose | Mary | 001 |
| 1124 | Jessy | Samuel | 002 |
| 1125 | John | Williams | 003 |
| 1126 | Lily | Holmes | 004 |



# Database Constraints

- **SQL constraints** are used to specify rules for the data in a table.
- If there is any violation between the constraint and the data action, the action is aborted by the constraint.
- Constraints can be specified when the table is created (inside the CREATE TABLE statement) or after the table is created (inside the ALTER TABLE statement).

- **Constraints can be defined in two ways**
  - 1. The constraints can be specified immediately after the column definition. This is called column- level definition.
  - 2. The constraints can be specified after all the columns are defined. This is called table-level definition.

# Database Constraints

There are 7 types of constraints that are grouped in 4 categories:

Entity Integrity:

1. Primary Key -- Identifies a row uniquely which will not allow NULL values and Duplicate Values
2. Unique Key -- Identifies a row uniquely which will allow NULL values

Referential Integrity:

3. Foreign Key -- References primary key of same table or another table

Domain Integrity:

4. Default -- Allow to set default value
5. Check -- Defines a logic on a column
6. Not Null -- Enforces the column must contain a value

User Defined Integrity:

7. Rules -- Based on the scenario a DBA can set own constraints

# Constraints

- CREATE TABLE employee
  ( id number(5),
  name char(20),
  dept char(10),
  age number(2),
  salary number(10),
  location char(10),
  CONSTRAINT emp_id_pk PRIMARY KEY (id)
  );

- CREATE TABLE employee
  ( id number(5), NOT NULL,
  name char(20),
  dept char(10),
  age number(2),
  salary number(10),
  location char(10),
  ALTER TABLE employee ADD CONSTRAINT PK_EMPLOYEE_ID PRIMARY KEY (id)
  );

# Constraints

NOT NULL

This constraint ensures all rows in the table contain a definite value for the column which is specified as not null. Which means a null value is not allowed.

The NOT NULL constraint enforces a field to always contain a value. This means that you cannot insert a new record, or update a record without adding a value to this field.

# Constraints

UNIQUE KEY

This constraint ensures that a column or a group of columns in each row have a distinct value. A column(s) can have a null value but the values cannot be duplicated.

- CREATE TABLE employee
  ( id number(5) PRIMARY KEY,
  name char(20),
  dept char(10),
  age number(2),
  salary number(10),
  location char(10) UNIQUE
  );

# Constraints

- CREATE TABLE employee
  ( id number(5) PRIMARY KEY,
  name char(20),
  dept char(10),
  age number(2),
  salary number(10),
  location char(10) CONSTRAINT loc_un UNIQUE
  );
- CREATE TABLE employee
  ( id number(5) PRIMARY KEY,
  name char(20),
  dept char(10),
  age number(2),
  salary number(10),
  location char(10),
  CONSTRAINT loc_un UNIQUE(location)
  );

# Constraints

PRIMARY KEY

This constraint defines a column or combination of columns which uniquely identifies each row in the table and NOT NULL

- CREATE TABLE employee
  ( id number(5) PRIMARY KEY,
  name char(20),
  dept char(10),
  age number(2),
  salary number(10),
  location char(10)
  );

# Constraints

- CREATE TABLE employee
  ( id number(5),
  name char(20),
  dept char(10),
  age number(2),
  salary number(10),
  location char(10),
  CONSTRAINT emp_id_pk PRIMARY KEY (id)
  );

- CREATE TABLE employee
  ( id number(5), NOT NULL,
  name char(20),
  dept char(10),
  age number(2),
  salary number(10),
  location char(10),
  ALTER TABLE employee ADD CONSTRAINT PK_EMPLOYEE_ID PRIMARY KEY (id)
  );

# Constraints

FOREIGN KEY

This constraint identifies any column referencing the PRIMARY KEY in another table. It establishes a relationship between two columns in the same table or between different tables. For a column to be defined as a Foreign Key, it should be a defined as a Primary Key in the table which it is referring. One or more columns can be defined as Foreign key.

- CREATE TABLE product
  ( product_id number(5) CONSTRAINT pd_id_pk PRIMARY KEY,
  product_name char(20),
  supplier_name char(20),
  unit_price number(10)
  );

# Constraints

```
CREATE TABLE order_items
( order_id number(5) CONSTRAINT od_id_pk PRIMARY KEY,
product_id number(5) CONSTRAINT pd_id_fk REFERENCES, product(product_id),
product_name char(20),
supplier_name char(20),
unit_price number(10)
);
```

- ```
  CREATE TABLE order_items
  ( order_id number(5) ,
  product_id number(5),
  product_name char(20),
  supplier_name char(20),
  unit_price number(10)
  CONSTRAINT od_id_pk PRIMARY KEY(order_id),
  CONSTRAINT pd_id_fk FOREIGN KEY(product_id) REFERENCES product(product_id)
  );
  ```
  ```

# Constraints

CHECK CONSTRAINT

- This constraint defines a business rule on a column. All the rows must satisfy this rule. The constraint can be applied for a single column or a group of columns.
- The CHECK constraint is used to limit the value range that can be placed in a column
- CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

- CREATE TABLE employee
  ( id number(5) PRIMARY KEY,
  name char(20),
  dept char(10),
  age number(2),
  gender char(1) CHECK (gender in ('M','F')),

  age int CHECK(age >0 and age <150),
  salary number(10),
  location char(10)
  );

# Constraints

- CREATE TABLE employee
  ( id number(5) PRIMARY KEY,
  name char(20),
  dept char(10),
  age number(2),
  gender char(1),
  salary number(10),
  location char(10),
  CONSTRAINT gender_ck CHECK (gender in ('M','F')) ,

    CONSTRAINT age_Ck CHECK )age int CHECK(age >0 and age <150)
  );

# Constraints

DEFAULT CONSTRAINT

- Specifies a default value for a column
- The DEFAULT constraint is used to insert a default value into a column. The default value will be added to all new records, if no other value is specified.

- ```
  CREATE TABLE Persons
  (
  P_Id int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Address varchar(255),
  City varchar(255) DEFAULT 'Sandnes'
  )
  ```
- ```
  CREATE TABLE Orders
  (
  O_Id int NOT NULL,
  OrderNo int NOT NULL,
  P_Id int,
  OrderDate date DEFAULT GETDATE()
  )
  ```

# Constraints

```
CREATE TABLE Orders
(
O_Id int NOT NULL,
OrderNo int NOT NULL,
P_Id int,
OrderDate date,
DEFAULT GETDATE() FOR Orderdate
```

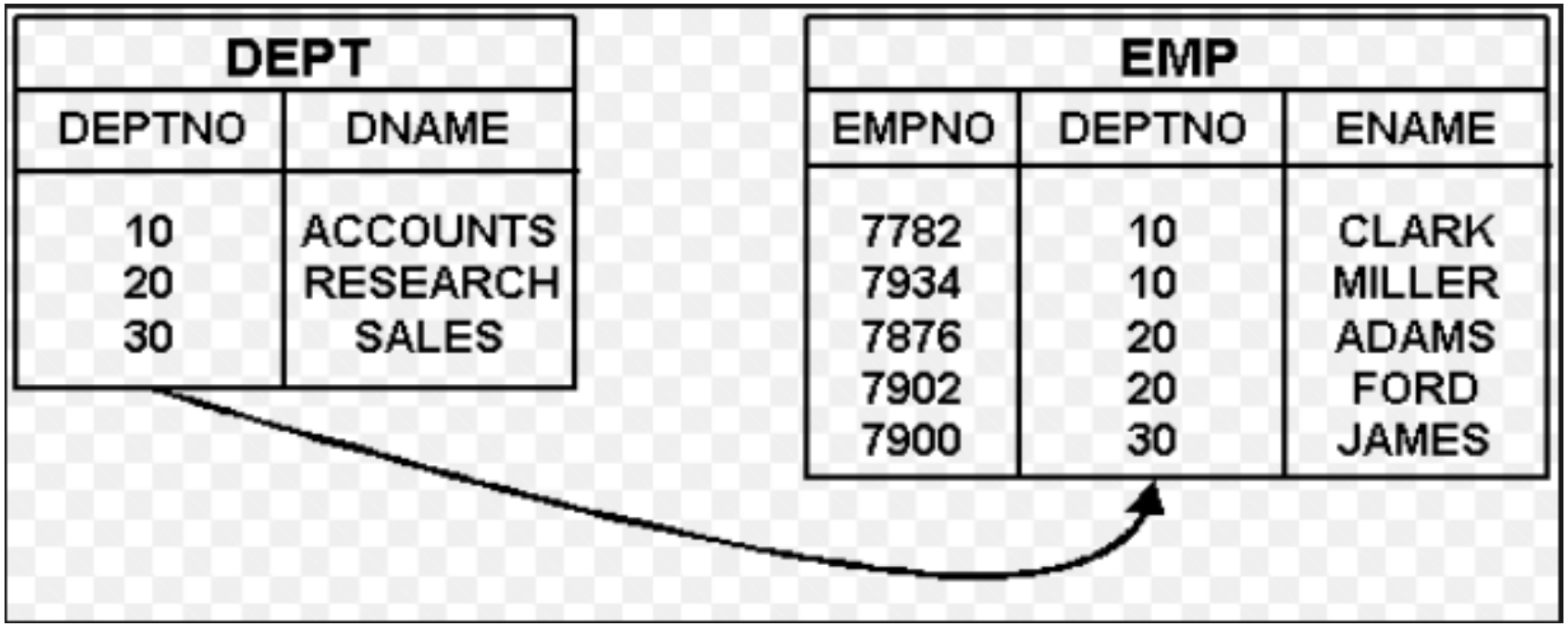
# Logical Relational Data Model

- A database model is a type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized and manipulated.
- The most popular example of a database model is the relational model, which uses a table-based format.
- **Logical**—this is the entity **relationship model** or Entity **Relationship** Diagram (ERD), and comprises entities, attributes, and relationships.
- **Relational**—this is the schema or **database model** and is comprised of tables, columns, views, and constraints.

# Referential integrity

What is Referential integrity?

**Referential integrity** (RI) is a relational **database** concept, which states that table relationships must always be consistent. In other words, Referential integrity means that the foreign key in any referencing table must always refer to a valid row in the referenced table. Referential integrity ensures that the relationship between two tables remains synchronized during updates and deletes.



**DEPT**

| DEPTNO | DNAME |
|---|---|
| 10 | ACCOUNTS |
| 20 | RESEARCH |
| 30 | SALES |

**EMP**

| EMPNO | DEPTNO | ENAME |
|---|---|---|
| 7782 | 10 | CLARK |
| 7934 | 10 | MILLER |
| 7876 | 20 | ADAMS |
| 7902 | 20 | FORD |
| 7900 | 30 | JAMES |

# Normalization

Database normalization (or normalization) is the process of organizing the columns (attributes) and tables (relations) of a relational database to minimize data redundancy.

| UNF | 1NF | 2NF | 3NF |
| --- | --- | --- | --- |
| • All the attributes of the database are simply listed, without any sense of order or grouping. | • It has an identifying key and it contains no repeating groups of data. | • It is in 1NF<br>• All non-key attributes are functionally dependent on the whole key (not part of the key) | • It is in 2NF<br>• It contains no transitive dependencies |

# UnNormalized data

| Enrollment Number | Student Name | Subject |
|---|---|---|
| 101 | Adam | Computer Science, Math |
| 102 | James | Computer Science, English |
| 103 | Paul | Social Science |
| 104 | Stephen | Computer Science, Math |

# First Normal Form-Normalization

- First Normal Form
  - Rule 1: Each Column should contain atomic values
  - Rule 2: Columns should contain values that are of the same type
  - Rule 3: The Column name should be unique

| Enrollment Number | Student Name | Subject |
|---|---|---|
| 101 | Adam | Computer Science |
| 101 | Adam | Math |
| 102 | James | English |
| 102 | James | Computer Science |
| 103 | Paul | Social Science |
| 104 | Stephen | Computer Science |
| 104 | Stephen | Math |

# Second Normal Form-Normalization

Second Normal Form : Should be in First Normal Form

Should not have any Partial Dependency

| Student Id (P) | Enrollment No | Student Name |
|---|---|---|
| 1 | 101 | Adam |
| 2 | 102 | James |
| 3 | 103 | Paul |
| 4 | 104 | Stephen |

| Subject Id | Subject Name |
|---|---|
| 01 | Computer Science |
| 02 | Math |
| 03 | English |
| 04 | Social Science |

| Score Id | Student Id | Subject Id | Marks | Trainers |
|---|---|---|---|---|
| 001C | 1 | 01 | 82 | Mr. J |
| 002M | 1 | 02 | 90 | Miss. M |
| 003E | 3 | 03 | 95 | Mr. E |
| 004S | 2 | 04 | 94 | Miss. S |

# Third Normal Form

Third Normal Form:  The table should be in 2 NF and should not have Transitive Dependency

| Score Id | Student Id | Subject Id | Marks | Exam Name | Total Marks |
|---|---|---|---|---|---|
| 001C | 1 | 01 | 82 | Computer Science Practical | 50 |
| 002M | 1 | 02 | 90 | English | 100 |
| 003E | 3 | 03 | 95 | Math | 100 |
| 004S | 2 | 04 | 94 | Social Science | 100 |

# Normalization (Example)

## Unnormalised Form

| Project Code | Project Title | Project Manager | Project Budget | Employee No. | Employee Name | Department No. | Department Name | Hourly Rate |
|---|---|---|---|---|---|---|---|---|
| PC010 | Pensions System | M Phillips | 24500 | S10001 | A Smith | L004 | IT | 22.00 |
| PC010 | Pensions System | M Phillips | 24500 | S10030 | L Jones | L023 | Pensions | 18.50 |
| PC010 | Pensions System | M Phillips | 24500 | S21010 | P Lewis | L004 | IT | 21.00 |
| PC045 | Salaries System | H Martin | 17400 | S10010 | B Jones | L004 | IT | 21.75 |
| PC045 | Salaries System | H Martin | 17400 | S10001 | A Smith | L004 | IT | 18.00 |
| PC045 | Salaries System | H Martin | 17400 | S31002 | T Gilbert | L028 | Database | 25.50 |
| PC045 | Salaries System | H Martin | 17400 | S13210 | W Richards | L008 | Salary | 17.00 |
| PC064 | HR System | K Lewis | 12250 | S31002 | T Gilbert | L028 | Database | 23.25 |
| PC064 | HR System | K Lewis | 12250 | S21010 | P Lewis | L004 | IT | 17.50 |
| PC064 | HR System | K Lewis | 12250 | S10034 | B James | L009 | HR | 16.50 |
| PC010 | Pensions System | M Phillips | 24500 | S10001 | A Smith | L004 | IT | 22.00 |
| | | | | S10030 | L Jones | L023 | Pensions | 18.50 |
| | | | | S21010 | P Lewis | L004 | IT | 21.00 |
| PC045 | Salaries System | H Martin | 17400 | S10010 | B Jones | L004 | IT | 21.75 |
| | | | | S10001 | A Smith | L004 | IT | 18.00 |
| | | | | S31002 | T Gilbert | L028 | Database | 25.50 |
| | | | | S13210 | W Richards | L008 | Salary | 17.00 |
| PC064 | HR System | K Lewis | 12250 | S31002 | T Gilbert | L028 | Database | 23.25 |
| | | | | S21010 | P Lewis | L004 | IT | 17.50 |
| | | | | S10034 | B James | L009 | HR | 16.50 |

# First Normal Form-Normalization

- First Normal Form(1NF)
  - The rule is: remove any repeating attributes to a new table. The process is as follows:

| Project Code | Project Title | Project Manager | Project Budget |
|---|---|---|---|
| PC010 | Pensions System | M Phillips | 24500 |
| PC045 | Salaries System | H Martin | 17400 |
| PC064 | HR System | K Lewis | 12250 |

| Project Code | Employee No. | Employee Name | Department No. | Department Name | Hourly Rate |
|---|---|---|---|---|---|
| PC010 | S10001 | A Smith | L004 | IT | 22.00 |
| PC010 | S10030 | L Jones | L023 | Pensions | 18.50 |
| PC010 | S21010 | P Lewis | L004 | IT | 21.00 |
| PC045 | S10010 | B Jones | L004 | IT | 21.75 |
| PC045 | S10001 | A Smith | L004 | IT | 18.00 |
| PC045 | S31002 | T Gilbert | L028 | Database | 25.50 |
| PC045 | S13210 | W Richards | L008 | Salary | 17.00 |
| PC064 | S31002 | T Gilbert | L028 | Database | 23.25 |
| PC064 | S21010 | P Lewis | L004 | IT | 17.50 |
| PC064 | S10034 | B James | L009 | HR | 16.50 |

# Second Normal Form-Normalization

- Second Normal Form(2NF)
  - The rule is: remove any non-key attributes that only depend on part of the table key to a new table.

| Project Code | Project Title | Project Manager | Project Budget |
|---|---|---|---|
| PC010 | Pensions System | M Phillips | 24500 |
| PC045 | Salaries System | H Martin | 17400 |
| PC064 | HR System | K Lewis | 12250 |

| Project Code | Employee No. | Hourly Rate |
|---|---|---|
| PC010 | S10001 | 22.00 |
| PC010 | S10030 | 18.50 |
| PC010 | S21010 | 21.00 |
| PC045 | S10010 | 21.75 |
| PC045 | S10001 | 18.00 |
| PC045 | S31002 | 25.50 |
| PC045 | S13210 | 17.00 |
| PC064 | S31002 | 23.25 |
| PC064 | S21010 | 17.50 |
| PC064 | S10034 | 16.50 |

| Employee No. | Employee Name | Department No. | Department Name |
|---|---|---|---|
| S10001 | A Smith | L004 | IT |
| S10030 | L Jones | L023 | Pensions |
| S21010 | P Lewis | L004 | IT |
| S10010 | B Jones | L004 | IT |
| S31002 | T Gilbert | L028 | Database |
| S13210 | W Richards | L008 | Salary |
| S10034 | B James | L009 | HR |

**2NF: Partial Key Dependencies Removed**

# Third Normal Form-Normalization

- Third Normal Form(3NF)
  - The rule is: remove to a new table any non-key attributes that are more dependent on other non-key attributes than the table key

**Project**

| Project Code | Project Title | Project Manager | Project Budget |
|---|---|---|---|
| PC010 | Pensions System | M Phillips | 24500 |
| PC045 | Salaries System | H Martin | 17400 |
| PC064 | HR System | K Lewis | 12250 |

**Project Team**

| Project Code | Employee No. | Hourly Rate |
|---|---|---|
| PC010 | S10001 | 22.00 |
| PC010 | S10030 | 18.50 |
| PC010 | S21010 | 21.00 |
| PC045 | S10010 | 21.75 |
| PC045 | S10001 | 18.00 |
| PC045 | S31002 | 25.50 |
| PC045 | S13210 | 17.00 |
| PC064 | S31002 | 23.25 |
| PC064 | S21010 | 17.50 |
| PC064 | S10034 | 16.50 |

**Employee**

| Employee No. | Employee Name | Department No. * |
|---|---|---|
| S10001 | A Smith | L004 |
| S10030 | L Jones | L023 |
| S21010 | P Lewis | L004 |
| S10010 | B Jones | L004 |
| S31002 | T Gilbert | L023 |
| S13210 | W Richards | L008 |
| S10034 | B James | L0009 |

| Department No. | Department Name |
|---|---|
| L004 | IT |
| L023 | Pensions |
| L028 | Database |
| L008 | Salary |
| L009 | HR |

**Department**

**3NF: Non-Key Dependencies Removed**

# Fundamental building blocks

Table: A table can store a mass Information in a tabular form

Fields: The information of a table stored against some heads, those are fields or columns. Columns shows vertically in a table.

Column Names: Each fields or column has an individual name. A table cannot contain same name of two different columns

Record: All the columns in a table makes a row. Each row contain all the information of an individual topics.

Column Value: Value of each field makes a row is the column value.

Key Field (Primary key and Foreign key): Each table should contain a field which can create a link with another one or more table is the key field of a table.

# SQL Server Versions

- Enterprise
  - SQL Server Enterprise delivers comprehensive high-end datacenter capabilities for demanding database and business intelligence requirements.
- Business Intelligence
  - SQL Server Business Intelligence empowers organizations to build and deploy secure, scalable and manageable self-service corporate BI solutions.
- Standard
  - SQL Server Standard provides core data management and business intelligence capabilities for non-critical workloads with minimal IT resources.
- Express
  - SQL Server Express is a free edition of SQL Server ideal for developing and powering desktop, web and small server applications.

# SQL Server Releases

- SQL Server 2017
- SQL Server 2016
- SQL Server 2014 SP1 12.0.4100.1
- SQL Server 2012 Sp1 11.0.3000.0 Sp211.0.5058.0
- SQL Server 2008 R2
- SQL Server 2008
- And so on 2005,2000,7.0
- **How to: Install SQL Server Express:**
  https://technet.microsoft.com/en-us/library/ms143722%28v=sql.90%29.aspx

# Installing SQL Server Express

- To begin installation, double-click sqlexpr.exe.
- On the End User License Agreement page, read the license agreement, and then select **I accept the licensing terms and conditions** check box. Click **Next**.
- The Welcome to the Microsoft SQL Server Installation Wizard page appears. Click **Next**.
- On the System Configuration Check page, the computer is scanned for potential installation problems. To interrupt the scan, click **Stop**. To proceed with setup after the scan completes, click **Continue**.
- On the Registration Information page, enter information in the **Name** and **Company** text boxes. Click **Next**.
- On the Feature selection page, select the program features to install and click **Next**.
- On the Instance Name page, select a **Default instance** or a **Named instance** for your installation. If you select **Default instance**, an existing default instance will be upgraded. If you select **Named Instance**, specify an instance name or the default named instance of **SQLExpress** is used. Click **Next**.
- The **Service Account** page lets you assign local system account or a domain user account to SQL Server service. This page does not appear if you select the **Hide advanced configuration options** check box

# Installing SQL Server Express

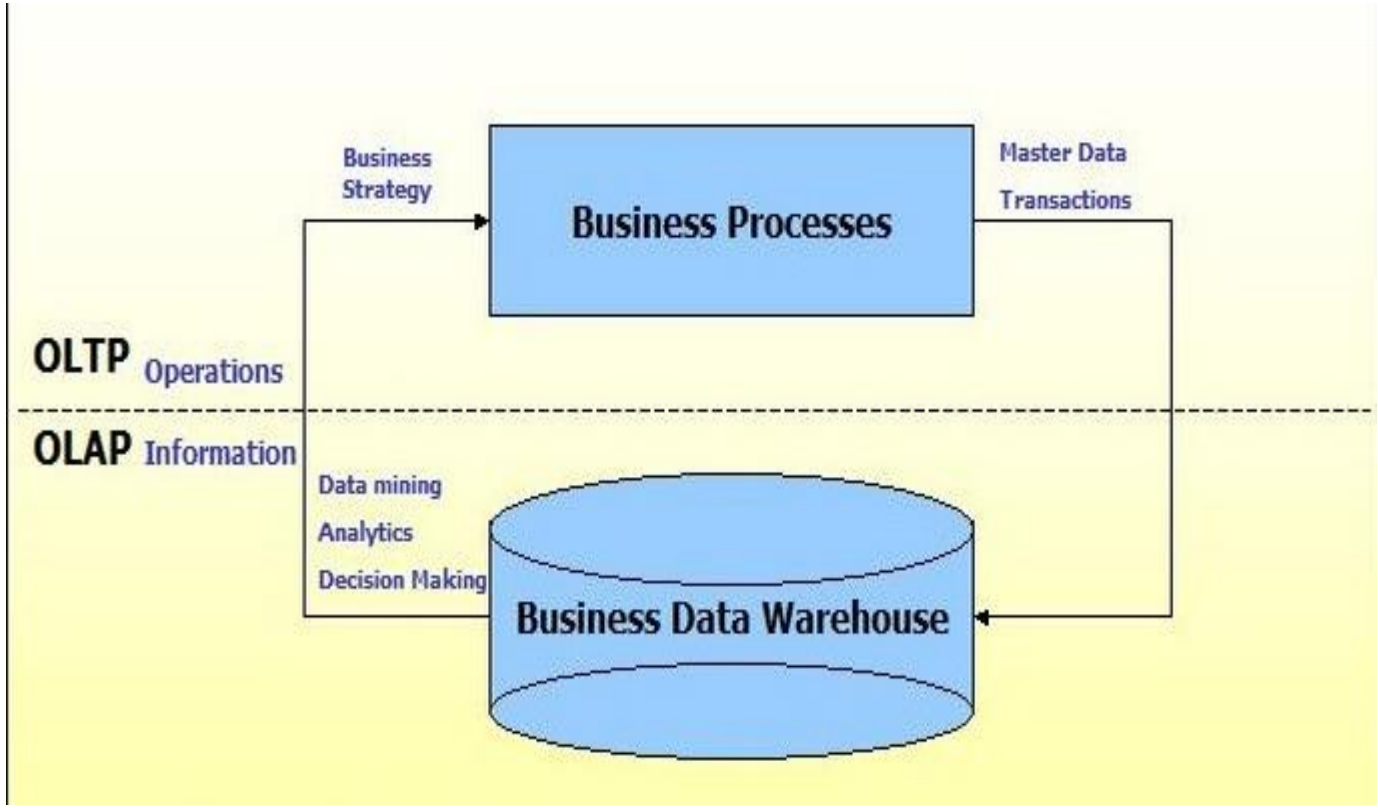
- On the **Authentication Mode** page, specify the security mode used to connect to your instance of SQL Server. If you select Windows Authentication, Setup creates an sa account that is disabled by default. Click **Next**.
- On the **Collations Settings** page, change the default **Collation designator and sort order** only if you must match collation settings in another instance of SQL Server or in another computer. Select **SQL collations** to match the sort order settings in earlier versions of SQL Server. Click **Next**. This page does not appear if you select the **Hide advanced configuration options** check box.
- On the **User Instances** page, specify if you want to generate a separate instance of the Database Engine for non-administrator users. By default, user instance functionality is enabled. To turn off user instances, clear the check box. This page does not appear if you select the **Hide advanced configuration options** check box. Click **Next**.
- On the **Error and Usage Report Setting** page, select whether to turn on error reporting and usage reporting for SQL Server and its components. By default, error reporting is turned on. To turn off error reporting, clear the check box. Click **Next**.
- On the **Ready to Install** page, click **Install** to finish installing SQL Server.
- On the **Setup Progress** page, you can monitor installation progress as setup proceeds. Click **Next**.
- On the **Completing the Microsoft SQL Server Installation Wizard** page, you can view the setup summary log by clicking the link provided on this page. To complete the SQL Server Installation Wizard, click **Finish**.

# Transactional vs Analytical database

- A **transactional database** is a database that allows for **CRUD** actions (Create, Read, Update, Delete) to be taken in a transaction way to make atomic changes.

- An **analytical database** is a database of data which is generally **read-only** and used to store current and historical data for the purpose of mining, generating statistics, projections, and otherwise analyzing it for a certain pattern or criteria.

# Transactional vs Analytical database



# Transactional vs Analytical database

| | OLTP System<br>Online Transaction Processing<br>(Operational System) | OLAP System<br>Online Analytical Processing<br>(Data Warehouse) |
|---|---|---|
| Source of data | Operational data; OLTPs are the original source of the data. | Consolidation data; OLAP data comes from the various OLTP Databases |
| Purpose of data | To control and run fundamental business tasks | To help with planning, problem solving, and decision support |
| What the data | Reveals a snapshot of ongoing business processes | Multi-dimensional views of various kinds of business activities |
| Inserts and Updates | Short and fast inserts and updates initiated by end users | Periodic long-running batch jobs refresh the data |
| Queries | Relatively standardized and simple queries Returning relatively few records | Often complex queries involving aggregations |
| Processing Speed | Typically very fast | Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes |
| Space Requirements | Can be relatively small if historical data is archived | Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP |
| Database Design | Highly normalized with many tables | Typically de-normalized with fewer tables; use of star and/or snowflake schemas |
| Backup and Recovery | Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability | Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method |