Data Modeling and T-SQL

Meetings / Methodology

Meeting Types

JRD – Joint Requirement Definition: Define theory of what is needed with written explanations

Will Create a BRD (Business Required Document)

Typically a meeting among users and main developers or team leads

JAD – Joint Application Development: Devise pictorial view of needed application features

Will create FRD (Function Requirement Document)

Typically a meeting with Developers, project managers, and techs

Typical Process of project

Development -> Q&A (Testers) -> Staging/UAT (Sample Users) -> Production (Public)

Meetings do not always go one way. It is very possible to gather the business requirements in a project, go through analysis, and then begin design only to find out that the users do not like what is being developed. In this case, things would restart at the business requirements.

Methodology

Waterfall – Straight shot through the process, resulting is less user input

Agile – Do circles in process until users are satisfied with product

Scrum – Time limited meetings each day run by Scrum Master

Spiral – Mix of Scrum and Agile methods

Developing a Database

Types

OLTP – Online Transaction Processing

Is the current database in use for users, holds only current info or limited historical

OLAP – Online Analysis Processing

Separate database holding historical data, used for business decisions and trends

RDBMS & DBMS – Relational Database Management System

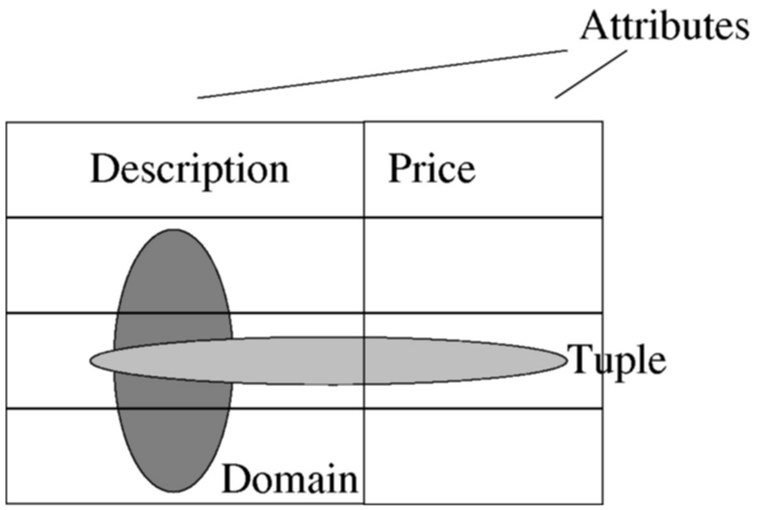
RDBMS – Oracle, SQL, Sybase, DB2

DBMS – Excel, Flat/Text files

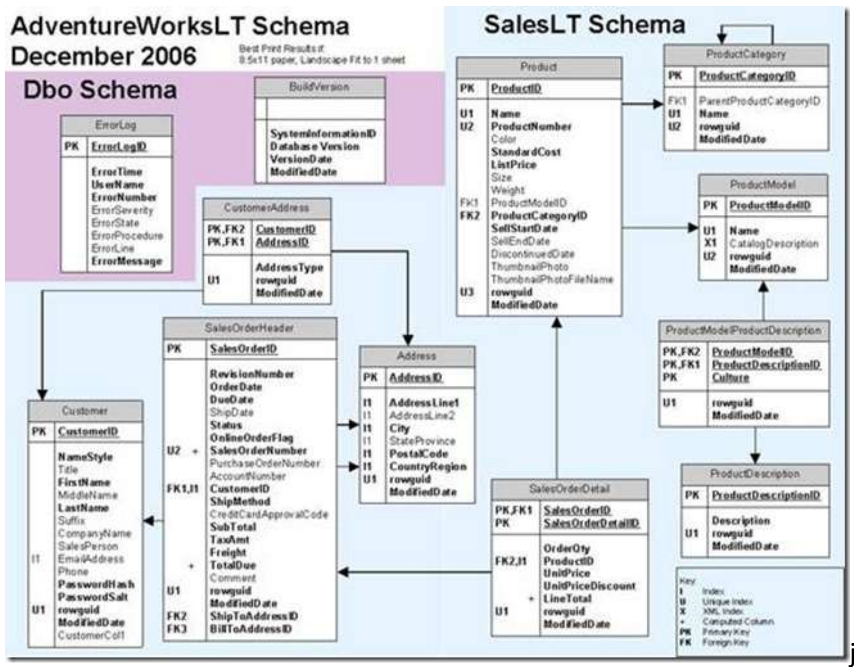
Data Modeling

Development of a database occurs to shape what will hold the data and how (Data Modeling)

Rows = Tuples / Columns Labels = Attributes (simple/composite/single/multi-valued) / Table = Entity (strong / weak) / Column Data = Domain



Schemas – The structure described in a formal language supported by the DBMS

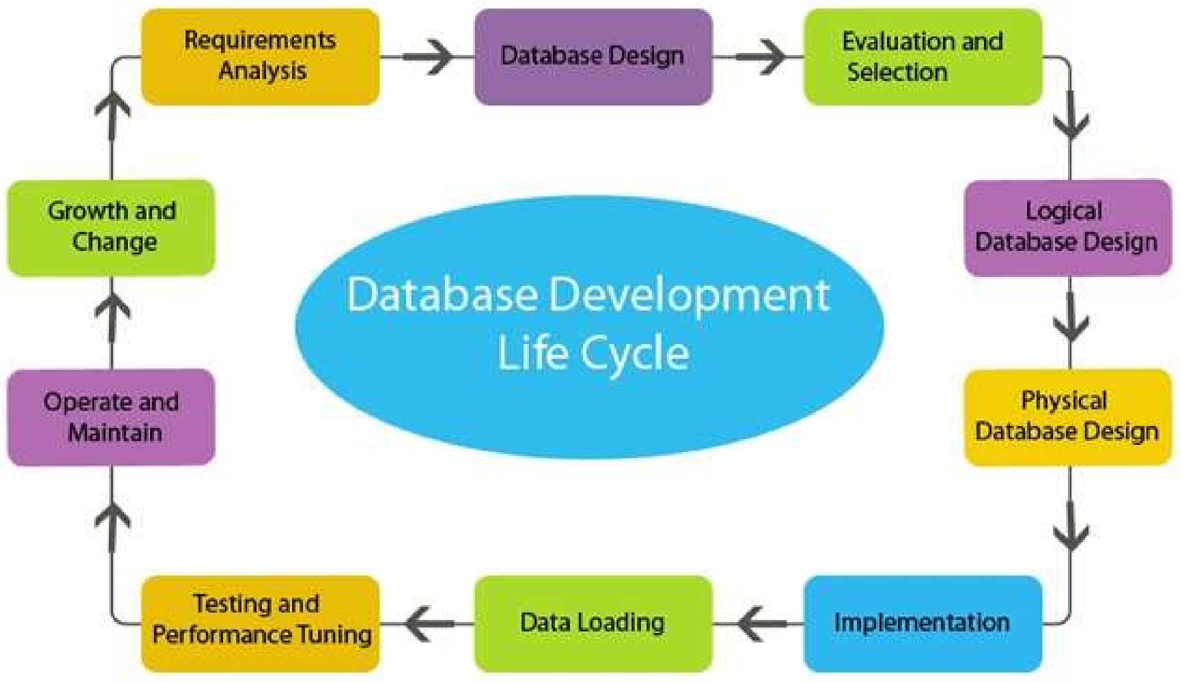


j

Concept Phase – Paper rough draft model, find relations

Logical Phase – Paper, develop constraints, normalization, ER Tool / Visio

Physical Phase – Implemented in server to be tested



Business Rules

Primary Key – Unique identifier row, no nulls, only one primary key allowed per table

Foreign Key – Key in another table that refers to the primary key of another

Unique Key – Used in U-NCI, allows 1 null value, 249 in 05 / 999 in 08 per table

Surrogate Key – Used in OLAP to play the role of primary key, automated sequence

Check – Verify data (type, range, etc.)

Default – A define value to fill in null areas

Nullability – Allowance of nulls

Relationships

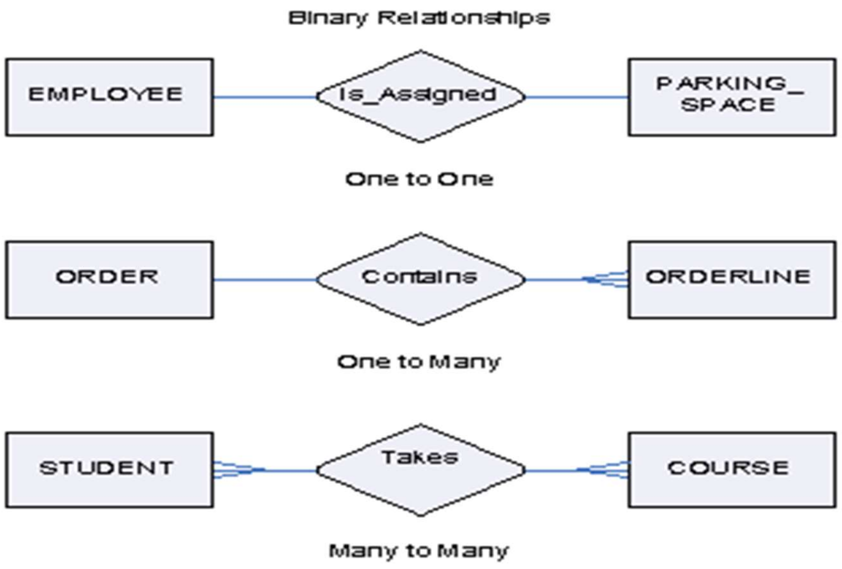
Relationships occur among entities

1 : 1 – One to One – Parent to child / Employee to spouse

1 : M – One to Many – Employee to Child

M : M – Many to Many – Broken into conjunction table with composite primary keys

1 : Fixed Cardinality – Set max/min relationships limit



Degree

Number of instances of entities in a relationship

Unary – Only one entity in relationship

Binary – Two entities in relationship

Ternary – Three entities in relationship

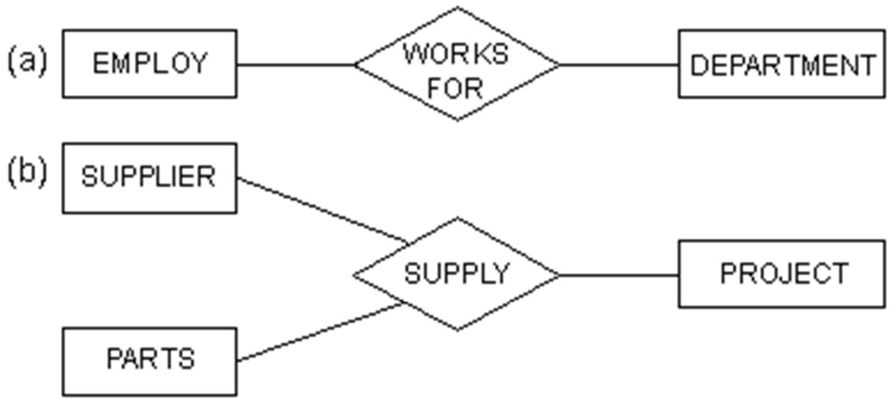


Figure 3.5 : (a) binary relationship WORKS_FOR
(b) ternary relationship SUPPLY

Domains

The set of all data types and ranges of values an attribute can assume

Allow the following:

1. Verify that the values for an attribute are valid

2. Ensure that various data manipulation operations are logical

3. Help conserve effort in describing attribute characteristics

Integrities

Entity Integrity*:*

Entity integrity involves the structure (primary key and its attributes) of the entity. If the primary key is unique and all attributes are scalar and fully dependent on the primary key, then the integrity of the entity is good. In the physical schema, the table's primary key enforces entity integrity.

Essentially, entity integrity is normalization.

Domain Integrity:

In relational theory terms, a domain is a set of possible values for an attribute, such as integers, bit values, or characters. Domain integrity ensures that only valid data is permitted in the attribute. Null ability (whether a null value is valid for an attribute) is also a part of domain integrity. In the physical schema, the data type and null ability of the row enforce domain integrity.

Referential Integrity:

A subset of domain integrity, referential integrity refers to the domain integrity of foreign keys. Domain integrity says that if an attribute has a value, then that value must be in the domain. In the case of the foreign key, the domain is the list of values in the related primary key. Referential integrity, therefore, is not an issue of the integrity of the primary key but of the foreign key.

The null ability of the column is a separate issue from referential integrity. It's perfectly acceptable for a foreign key column to allow nulls. Several methods of enforcing referential integrity at the physical-schema level exist. Within a physical schema, a foreign key can be enforced by declarative referential integrity (DRI) or by a custom trigger attached to the table.

User-Defined Integrity:

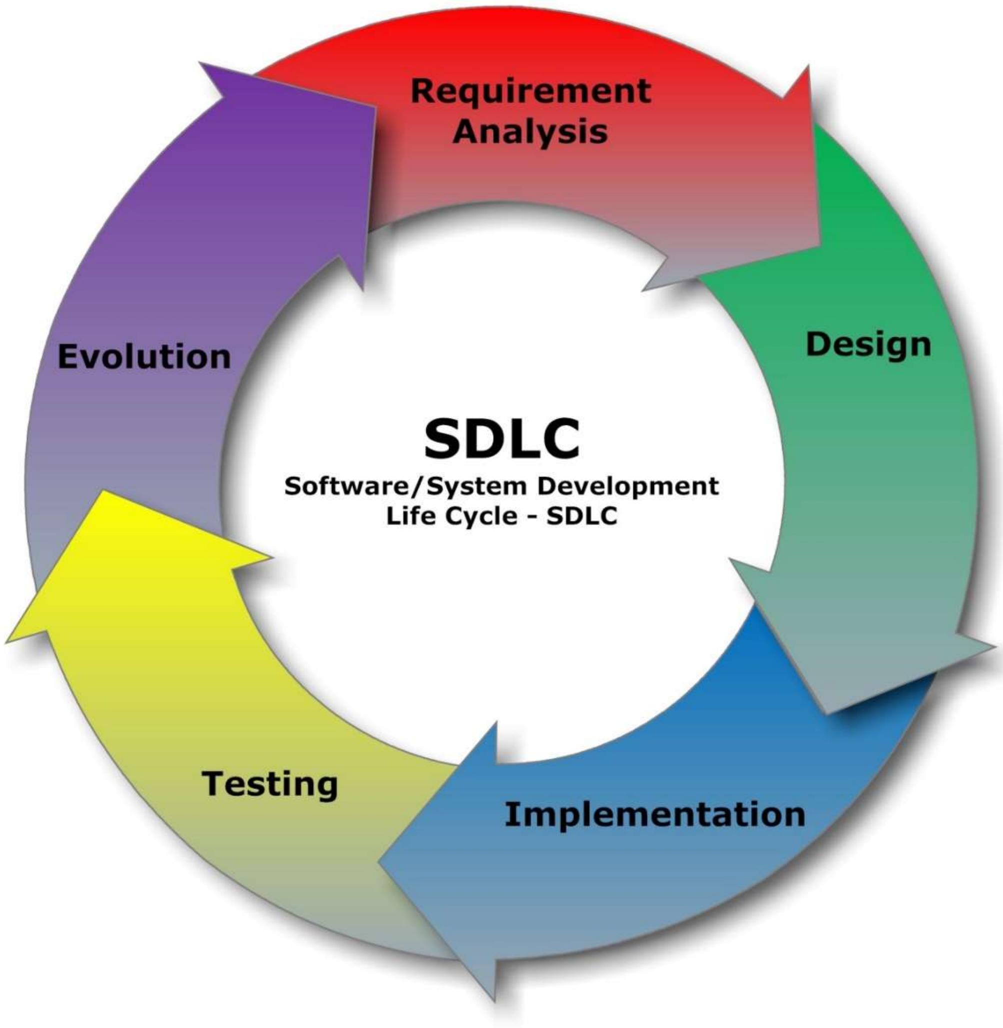
Besides the relational theory integrity concerns, the user-integrity requirements must also be enforced, as follows:

Simple business rules, such as a restriction to a domain, limit the list of valid data entries. Check constraints are commonly used to enforce these rules in the physical schema. Complex business rules limit the list of valid data based on some condition. For example, certain tours may require a medical waiver. Implementing these rules in the physical schema generally requires stored procedures or triggers.

Some data-integrity concerns can't be checked by constraints or triggers. Invalid, incomplete, or questionable data may pass all the standard data-integrity checks. For example, an order without any order detail rows is not a valid order, but no automatic method traps such an order. SQL queries can locate incomplete orders and help in identifying other less measurable data-integrity issues.

SDLC

The processes listed follow the order of the SDLC or Software Development Life Cycle



SQL Server

SQL – Structured Query Language: Common versions 2005, 2008, 2008 R2, 2012

Authentication

Two main methods to log into SQL Server

Windows Credentials – Common for local machines, uses window login info

SQL Server / Mixed – Uses Windows and a set login specified on the server for users

Database Objects

7 main: 6 permanent with 1 temporary

P – **Tables** – Naming schema <dbname>.<schemaname>.<tablename>

P – **Views** – Create a version of table with only needed data (filtered window)

P – **Store Procedures** – A set of SQL queries that are saved and can be called.

P – **User Defined Functions –** Similar to SP which different properties

P – **Triggers** – Act as constraints during certain events

P – **Indexes** – Map data and improve query performance

T – **Cursors** – Must be opened and closed, work as while statements

SQL Query Types

DDL – Data Definition Language – Create/Alter/Drop

DML – Data Modification Language – Insert/Update/Delete/Truncate

DQL – Data Query Language – Select, From, Where, Group, Having, Order

DCL – Data Control Language – Grant, Revoke, Deny

SQL Query Information / Examples

Data Types

INT – Integer (Sales ID int)
Float, Boolean, Big Int, Money
Varchar – Characters with allowable symbols. Max=2.1GB
Char – Characters only with no symbols. Max=8KB

Wild Cards

Used with Like and Not Like. Also have EXCEPT clause to find symbols

% , _ , [ ] , ^ (Data like this, empty space, contains these, not like this)

Select * From Employee Where Name LIKE ‘B%’

System Stored Procedures

Created by SQL Server when installing. Perform various tasks in SQL

sp_, sp_who, sp_who2, sp_helptxt, sp_database

Operators (Mathematical)

= - Equal to

< >, != - Is not equal to

<, >, >=, <= - Less than, greater, GT or Equal, LT or Equal

Logical Operators

These operators compare two or more conditions at a time to determine if a row can be selected for the output

OR – At least one of the conditions must be met

AND – All conditions must be met

NOT – Condition must be false for the row

```
Select first_name, last_name, subject
From student_details
Where subject = ‘Math’ or subject = ‘Science’
```

Set Operators

Set operators combine results from two or more queries into a single result set for use or filters based on another result set

Set operators are used with two select statements and in both statements we must have the same number of columns for each, and each column being combined must have the same metadata/data type. They must be in order for both select statements so the results sets can be combined.

Except – Returns distinct values from left query that are not found on the right

Intersect – Returns distinct values found in both queries

Union – Combines unique results from both queries into single result set

Union All – Combines all results from both queries, including duplicates

Aggregate Functions

Perform mathematical operations and must be used with Group By

Sum(), count(), max(), min(), avg()

Group by must be used whenever you are querying any non-aggregated columns, such as names, titles, etc. When group by is used, you can also include having if needed. If your query contains a having clause, it must also contain a group by.

Having is used to filter aggregation columns, much in the way that where can filter out most other results. Having is specifically used for aggregates to find or set conditions for the aggregated columns.

Example Query 1

```
Select sum(Salary), EID
From Employee
Where EID between 1 and 20
Group by EID
Having sum(Salary) > 100
Order by EID
```

Ctrl + R to hide results

String Concatenation

Creating a string with several different data bits from columns or tables

Fullname = FN + MN + LN → FirstMiddleLast (need spaces)

= FN + ‘ ‘ + MN + ‘ ‘ + LN → First Middle Last (but what if there is no middle name?)

= FN + ‘ ‘ + MN + ‘ ‘ + LN → First   Last (needs to be fixed)

= FN + ‘ ‘ + ISNULL(MN, ‘ ‘) + ‘ ‘ + LN → First Last (if MN, then insert, else add space)

Constraints

Specify certain criteria for data in the table

Not Null, Check, and Default are some examples. Not null will make say that no nulls can be entered into the data set. Check will check that the value in a column meets what is specified for criteria. Default will insert a default value into the space if one is not.

Rules

Create rules to keep data in check and prevent incorrect insertions of data. They are used to establish perhaps a constraint that is used repeatedly as a rule, and then simply bind them to tables or columns.

```
CREATE RULE range_rule
AS
@range>= $1000 AND @range <$20000

Exec sp_bindrule ‘range_rule’, HRE.EMP.Salary
```

Sub-String

A substring returns a sub or part of a string or statement. The expression is set as substring (expression, start, length). Pick the column, which character to start at, and then how far to go.

Stuff

A function that inserts a string into another string. It "stuffs" in data. The expression is set as STUFF(expression, start, length, replace expression)

Replace

Function that replaces all instances of a specified string or expression with another expression or string given. The expression is set as REPLACE(expression, pattern to find, replacement)

System Functions

Cast – Change data type (int to char, varchar to numeric, int to money)

Convert – Change data type and styling

Date Functions

Getdate() – System Date

Getutcdate() – Common Time

Datepart() – Grab a specific part of date

-**Year()** / **Month()** / **Day()**

Dateadd()

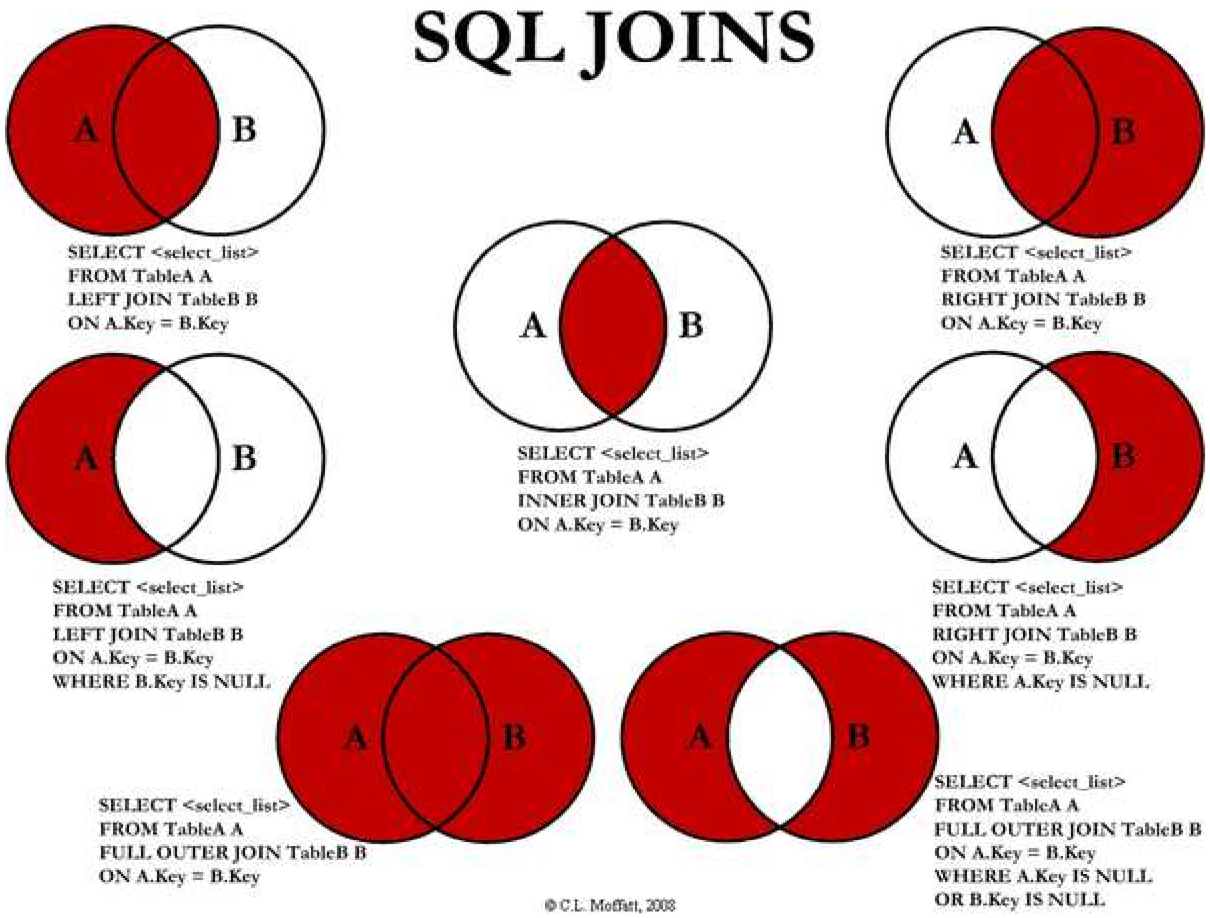
Datediff() – Difference between two dates

Working with Tables

Joins

Joins are used to combine tables based on a common column or data. There are three types of joins: Merge, Hash, and Loop. These are how the joins occur.

Inner Joins / Left Join / Left Outer Join / Right Join / Right Outer Join / Full Join / Self / Cross Join



Set Operators

Set operators join or filter results with either Union, Union All, Intersect, or Except. There are a few rules that must be followed in using set operators.

First, the number of columns called in the first select statement used must be the same as the number of columns called in the second statement.

Second, all columns used must have matching metadata or data types to be joined. With that being said, both select statements must call their columns in the same order so all data types match up and can be added.

Union

Union joins two or more select statement results. Each select statement must have the same number of columns used, since the data will be dumped into a table

There are two types of Union: Union and Union All

Union will display only distinct values

Union All will display all the data, even duplicates

```
Select column from table1
Union
Select column from table2
```

Except and Intersect

Except and intersect will have the same syntax as union, with a select statement and then either except or intersect, followed by a second select statement.

Except will get all the results that are not in the second select statement, while the intersect will get results that are in both.

Creating a Table

Using DDL operations, you can create, alter, or drop tables. The table you design will be tailored to fit the data you want. You can define the columns and the data type, as well as the limit of characters.

```
CREATE table Test01
(TESTID int,
TestName varchar(20),
TestAddress varchar(40))
```

Data can also be stored in temporary storage using an AS condition. This is not quite a temp table though, so it'll be gone when the query is done. `Select Top1 TEMP.SalesORderID From (Select Top 3 * from Sales.SalesOrderHeader) As Temp`

Views

Views are used as a filtered window to a table or a virtual table. They will display the data from a table based on what you choose.

Altering a view will only alter the base table referenced

An index can be created on a view, only if it is Schema bound

Views can be encrypted, schema bound, have with check constraint, and used to see metadata

Schema Binding means the view is directly connected to table and the data can't be altered at all until the bound is broke

With Check on view restricts the DML for users, preventing changes of table

```
Create view <view name> AS 'Alias'
Select <requested data>
From <tablename>
Where<constraints>
```

View Rules

1. Permanent Table Reference / NO TEMP

2. No Computed By
3. Order by allowed only w/Top clause
4. Into Options & Hints not allows
5. Schema – Three-part naming
6. Schema – No Select *

Rankings

Gives rankings to data based on the selected value. Data can either be broken up via the order of some column, or by partitioning using multiple columns. A rank must be followed by an OVER clause. That order clause must be followed by an ORDER BY clause, and may or may not have a PARTITION BY clause.

Rank – 1,1,3,3,5

Dense_Rank – 1,1,2,2,3

Ntile – Breaks up data in random order and ranking

Row_Number – 1,2,3,4,5 – ranking based on table location going down rows

```
Select Row_Number() OVER (Partition By ID, Name
                          Order By ID) as 'RwNo'
From Employee
```