MODULE 2

Key Terms

- Relation
  - Table in a database
- Attribute
  - Column in a relation (table)
- Tuple
  - Row in a relation (table)
- Candidate key
  - Means of uniquely identifying a tuple in a relation based on one or more of its attributes
- Primary key
  - One of the candidate keys selected to be primary key
- Surrogate key
  - Created when none of the candidate keys is deemed to be suitable (autonumber or identity columns commonly used)
  - Performance, convenience
  - Can still enforce uniqueness on candidate keys
- Foreign key
  - An attribute or group of attributes that contain primary key value of another relation

Relational Theory

- Tables are relations
- Columns are attributes
- Rows are tuples
- Tuples must be unique
  - Need to be able to find a tuple to work with it
  - Need a way to uniquely identify it based on one or more attributes
  - Any unique combination of attributes is a *candidate key*
  - One of the candidate keys becomes the primary key

Surrogate Keys   Identity Keys, etc.

- Should never be necessary in a perfect world
- Every tuple should have an attribute or set of attributes that uniquely identifies it
  - Reflect characteristics of entity in the real world
- If databases were perfect, wouldn't need surrogate keys
  - But are often used as a concession
- Foreign key attributes let you reference tuples in another relation

☐ Data Integrity
☑ Entity
☐ Domain
☐ Referential

Data Integrity

Data Integrity validates the data before getting stored in the columns of the table.
SQL Server supports four type of data integrity:

Entity Integrity

Entity Integrity can be enforced through indexes, UNIQUE constraints and PRIMARY KEY constraints.

Domain Integrity

Domain integrity validates data for a column of the table.

It can be enforced using:
Foreign key constraints,
Check constraints,
Default definitions
NOT NULL.

Referential Integrity

FOREIGN KEY and CHECK constraints are used to enforce Referential Integrity.

User-Defined Integrity

It enables you to create business logic which is not possible to develop using system constraints. You can use stored procedure, trigger and functions to create user-defined integrity.

1st Normal Form

☐ Breaking information **down** into the most atomic units necessary to solve the business Solution;

    ☐ Address Location separated into 2 fields is Over-Normalizing

        1001   Fair Way Rd.

      (Address Num and Address Street)

Not First Normal Form1

LABS\SQLEXPRESS0...o.Not-1st-Normal1

EmployeeID	Name
101	Mickey The Mouse
102	Donald Duck
NULL	NULL

First Normal Form1

LABS\SQLEXPRESS0...- dbo.1st-Normal1

EmployeeID	FirstName	MiddleName	LastName
101	Mickey	The	Mouse
102	Donald	NULL	Duck
NULL	NULL	NULL	NULL

Not First Normal Form2

LABS\SQLEXPRESS0...o.Not-1st-Normal2

EmployeeID	Project1	Project2
101	My Project	Your Project
102	My Project	
NULL	NULL	NULL

First Normal Form2

LABS\SQLEXPRESS0...- dbo.1st-Normal2

EmployeeID	Project
100	My Project
100	Your Project
101	My Project
NULL	NULL

2nd Normal Form

Second Normal Form

The database should be in first normal form, and every attribute should be functionally dependent on the entire primary key.

☐ Separation of Information to separate Table/Relations

Not Second Normal Form

EmployeeID and ProjectID are a composite Primary

	EmployeeID	ProjectID	Project
	101	201	My Project
	101	202	Your Project
▶*	NULL	NULL	NULL

LABS\SQLEXPRESS0...o.Not-2nd-Normal

Second Normal Form1

LABS\SQLEXPRESS0...dbo.2nd-Normal1

	EmployeeID	ProjectID
	101	201
	101	202
▶*	NULL	NULL

Second Normal Form2

LABS\SQLEXPRESS0...dbo.2nd-Normal2

	ProjectID	Description
	201	My Project
	202	Your Project
▶*	NULL	NULL

☐ Separation to Tables

99 %

Results   Messages

	EmployeeID	ProjectID	Description
1	101	201	My Project
2	101	202	Your Project

Third Normal Form

The database should be in second normal form, and no attribute should be functionally dependent on an attribute that isn't in the primary key.

In Correlation:

Don't store calculated values. Store the raw data, and use queries to perform calculations as needed.

☐ **Not Third Normal Form1**

LABS\SQLEXPRESS0....Not-3rd-Normal1   ⊟ ✕   LABS\SQLEXPRESS0....Not-3rd-Normal1

EmployeeID	ProjectID	Hours	Rate	Total
101	201	8	10	80
102	201	5	20	1000
NULL	NULL	NULL	NULL	NULL

☐ **Not Third Normal Form2**

LABS\SQLEXPRESS0....Not-3rd-Normal2   ⊟ ✕   LABS\SQLEXPRESS0....Not-3rd-Normal2

EmployeeID	Address	City	State	ZipCode
101	P.O. Box 1000	Sacramento	CA	98530
102	1021 Fort Street	Placerville	CA	95667
NULL	NULL	NULL	NULL	NULL

Third Normal Form2

EmployeeID	Address	ZipCode
101	P.O. Box 1000	98530
102	1021 Fort Street	95667
NULL	NULL	NULL

Primary Candidate Keys

Sunday, May 21, 2017   11:37 AM

File is: PrimaryCandidate.SQL

```sql
USE Schools;
GO
----Create Test Table for with default columns values
CREATE TABLE PrimaryCandidateKeys
(
    EmployeeID uniqueidentifier DEFAULT NEWID(),
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    PRIMARY KEY (EmployeeID)
)
GO

INSERT INTO PrimaryCandidateKeys(LastName, FirstName)
VALUES ('Mouse', 'Mickey');

Select * From PrimaryCandidateKeys

Drop Table PrimaryCandidateKeys;
GO
```

100 %

Results Messages

	EmployeeID	LastName	FirstName
1	DCB424DA-2F1F-44D6-A1DA-319524F08DF1	Mouse	Mickey

Primary Surrogate Keys

```
PrimarySurrogate.sq...r (LABS\dellp (58))   ×   DateTypes.sql - (lo...er (LABS\dellp (51))

 USE Schools;
 GO
 ----Create Test Table for with default columns values
 CREATE TABLE PrimarySurrogateKeys
 (
     EmployeeID int identity(1,1),
     LastName varchar(255) NOT NULL,
     FirstName varchar(255),
     PRIMARY KEY (EmployeeID)
 )
 GO

 INSERT INTO PrimarySurrogateKeys(LastName, FirstName)
 VALUES ('Mouse', 'Mickey');

 Select * From PrimarySurrogateKeys

 Drop Table PrimarySurrogateKeys;
 GO
```
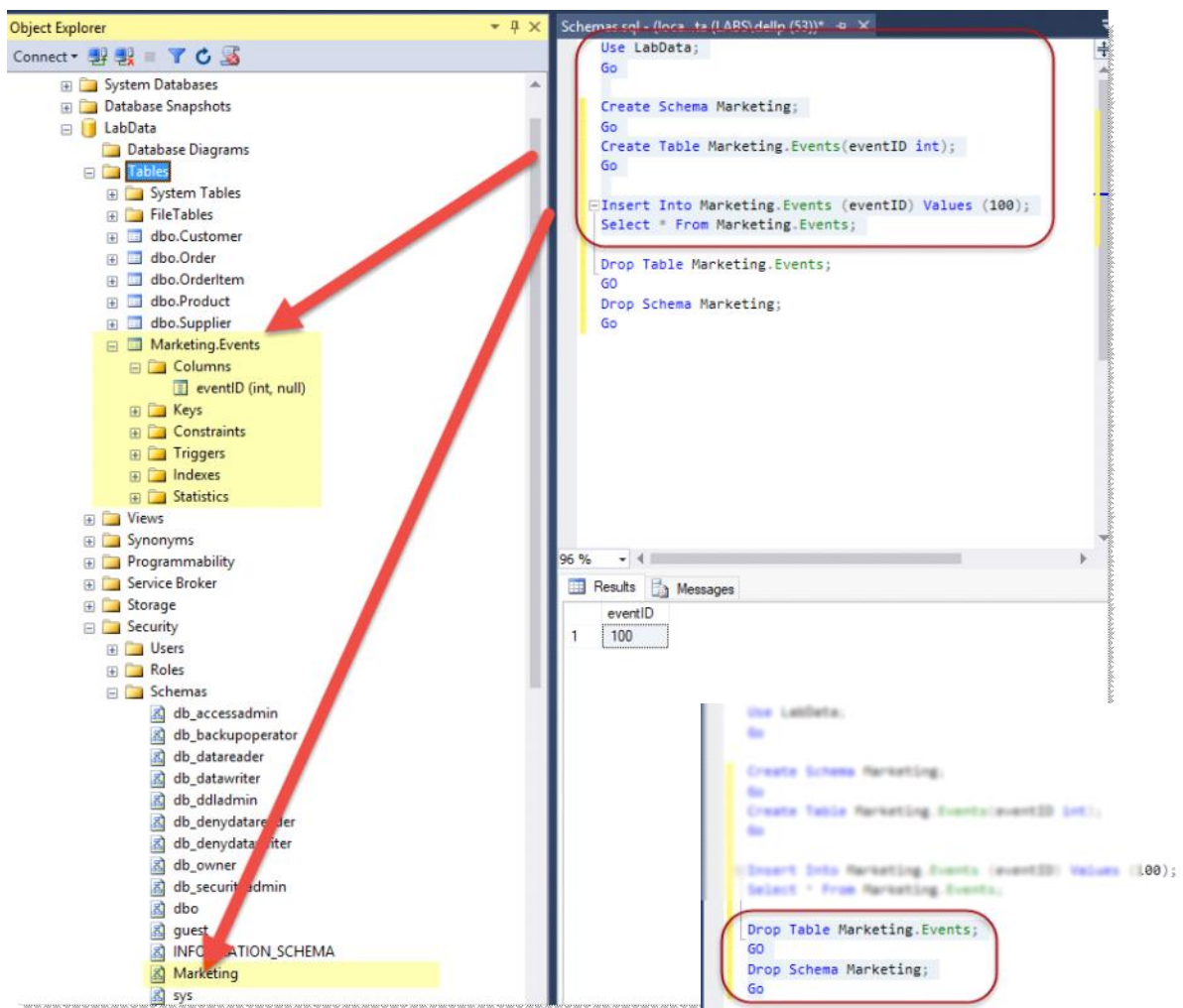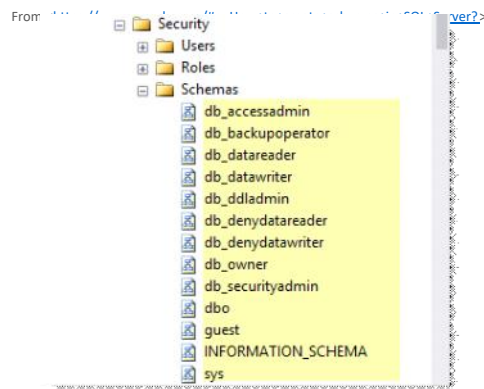
Schemas

<mark>File: Schemas.sql</mark>

A **schema** is a distinct namespace to facilitate the separation, management, and ownership of database objects.

From <https://....ver?>

<mark>Add Technical.Events to the Equation</mark>

Schema Object Transfer

Friday, May 26, 2017   6:30 PM

```sql
Use Schools;
Go

Create Schema Labs
Go

Create Table Demo
(
    ID Int Identity(1,1) Primary Key,
    Description varchar(30)
)

exec sp_tables;
Alter Schema Labs Transfer dbo.Demo;
Exec sp_tables;
```

166 %

Results | Messages

	TABLE_QUALIFIER	TABLE_OWNER	TABLE_NAME	TABLE_TYPE	REMARKS
1	Schools	dbo	Course	TABLE	NULL
2	Schools	dbo	CourseColumnStore	TABLE	NULL
3	Schools	dbo	CourseInMemory	TABLE	NULL
4	Schools	dbo	CourseInMemory_old	TABLE	NULL
5	Schools	dbo	Demo	TABLE	NULL
6	Schools	dbo	EmployeeClustered	TABLE	NULL
7	Schools	dbo	EmployeeNonClustered	TABLE	NULL
8	Schools	dbo	Person	TABLE	NULL
9	Schools	dbo	PersonHeap	TABLE	NULL
10	Schools	dbo	PersonStats	TABLE	NULL

	TABLE_QUALIFIER	TABLE_OWNER	TABLE_NAME	TABLE_TYPE	REMARKS
6	Schools	dbo	EmployeeNonClustered	TABLE	NULL
7	Schools	dbo	Person	TABLE	NULL
8	Schools	dbo	PersonHeap	TABLE	NULL
9	Schools	dbo	PersonStats	TABLE	NULL
10	Schools	Labs	Demo	TABLE	NULL
11	Schools	sys	trace_xe_action_map	TABLE	NULL

Query executed successfully.   (local)\sqlexpress01 (13.0 ... LABS\dellp (56) Schools 00:00:00 986 rows