



EPL342 –Databases

Lab 3

ER Modeling (Entities) in ERD+ Conceptual Modeling in SQL Server 2017



Before We Begin

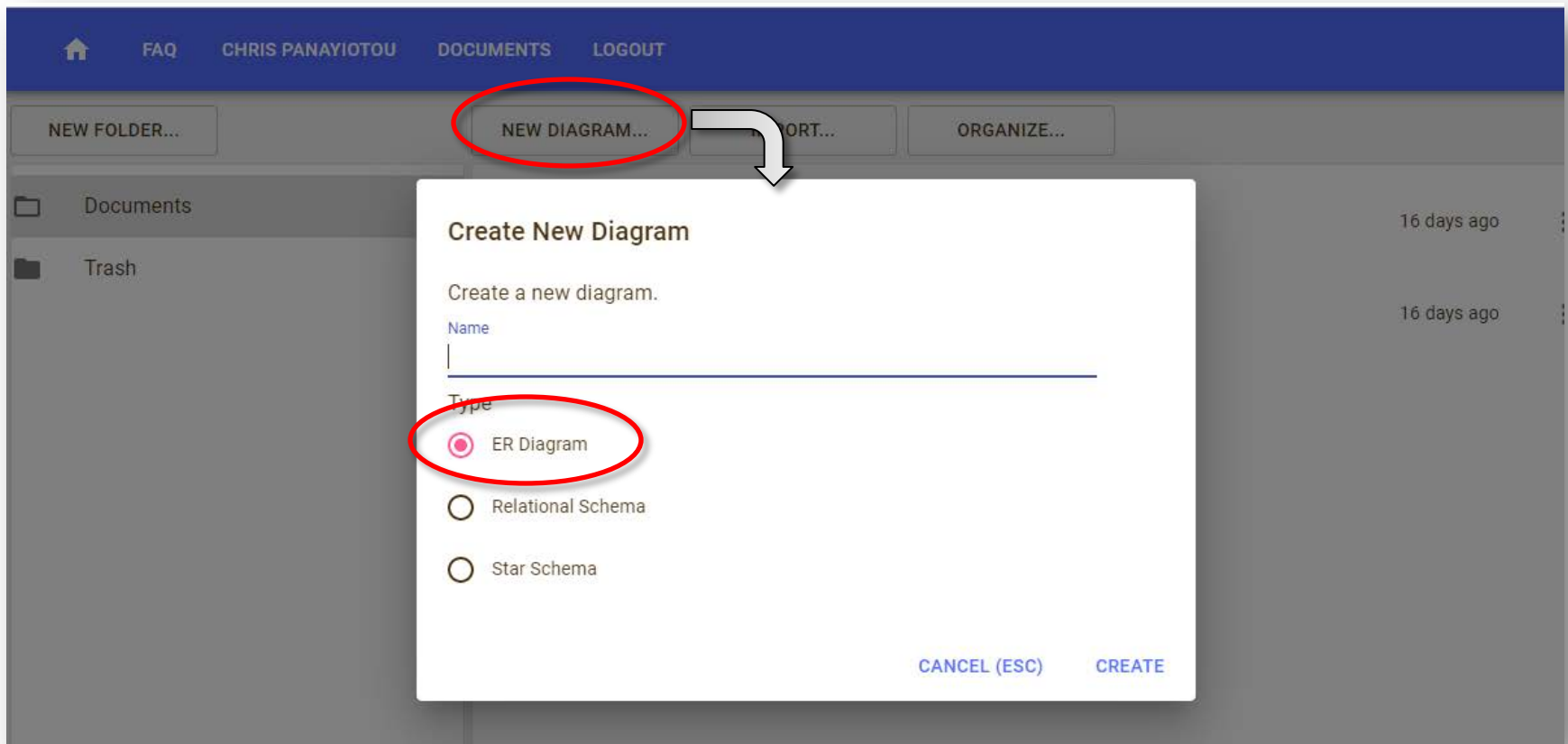
- Visit <https://erdplus.com/> and create an account
- Start the SQL Server Management Studio
 - Server: mssql.cs.ucy.ac.cy
 - Authentication: SQL Server Authentication
 - Username: <check your email>
 - Password: <check your email>

Lab Introduction



- COMPANY Database
- During your [lecture 4](#), you have identified 4 entities consisting of the COMPANY db:
 - DEPARTMENT
 - PROJECT
 - EMPLOYEE
 - DEPENDENT
- Our job is to design the entities/tables based on the requirements

ERD+ - Create a new Document



ERD+ - Create new Entity



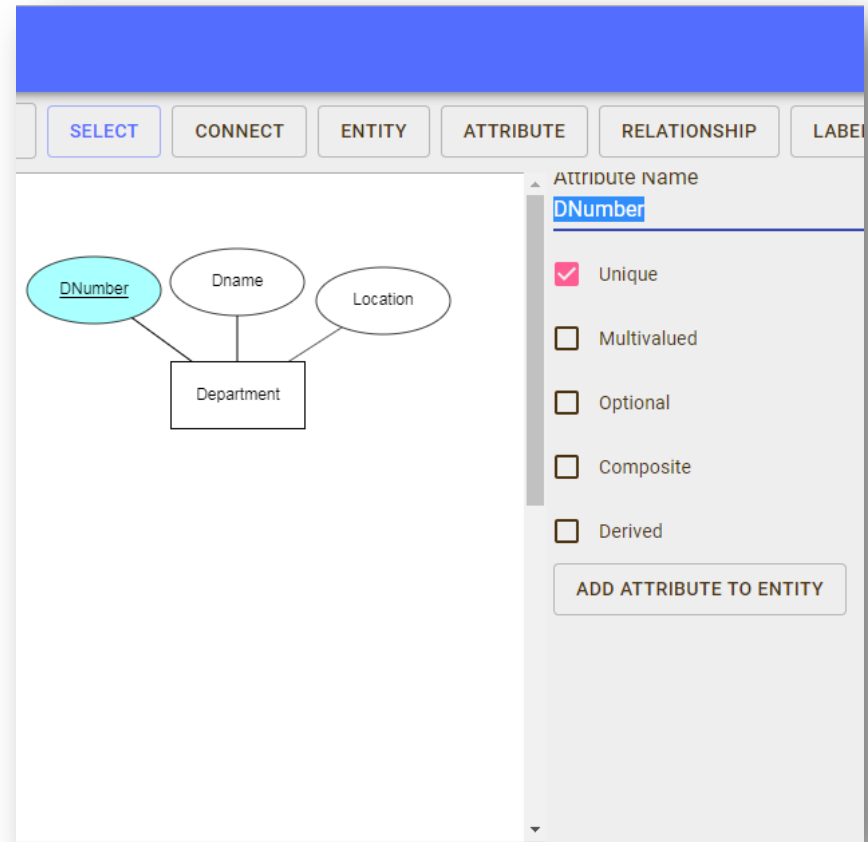
- To create a new entity open the new diagram and from the menu select entity

The screenshot displays the ERD+ software interface. At the top, a blue navigation bar contains a home icon, 'FAQ', 'CHRIS PANAYIOTOU', 'DOCUMENTS', and 'LOGOUT'. Below this is a toolbar with buttons for 'MENU', 'SAVE', 'UNDO', 'REDO', 'DELETE', 'SELECT', 'CONNECT', 'ENTITY', 'ATTRIBUTE', 'RELATIONSHIP', and 'LABEL'. The 'ENTITY' button is circled in red. A large grey arrow points from this button to a central diagram area. In the diagram area, a light blue oval labeled 'NewAttribute' is connected to a white rectangle labeled 'Department'. To the left of the diagram is a panel with 'Entity Name' set to 'Entity' and 'Type' set to 'Regular'. Below these options is a red-circled 'ADD ATTRIBUTE' button. A second large grey arrow points from this button to the right-hand panel. This panel shows 'Attribute Name' set to 'NewAttribute' and a list of checkboxes for 'Unique', 'Multivalued', 'Optional', 'Composite', and 'Derived'. At the bottom of this panel is an 'ADD ATTRIBUTE TO ENTITY' button.



ERD⁺ - Entity Properties

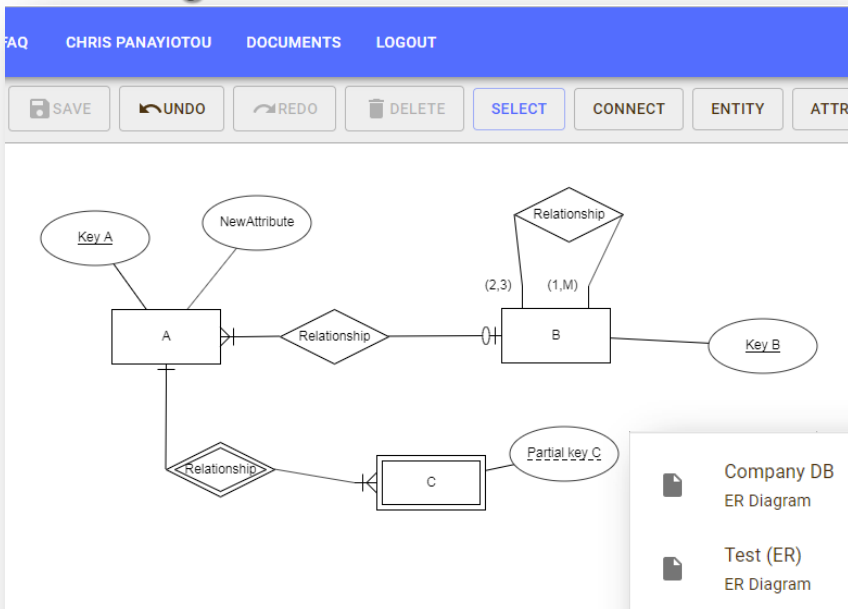
- Rename the newly created entity to DEPARTMENT
- Add to it the following attributes:
 - DNumber (Key)
 - DName
 - Location



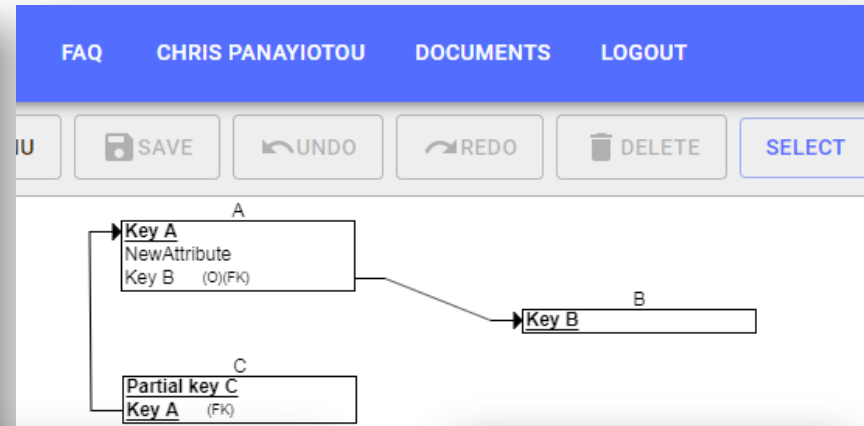
ERD+ - Different Documents Types



ER Diagram



Relational Schema



For Relational Schema documents you can also generate the corresponding SQL statements that create it!

The screenshot shows a document list with three items: 'Company DB ER Diagram' (16 minutes ago), 'Test (ER) ER Diagram' (9 minutes ago), and 'Test (Relational) Relational Schema'. A context menu is open over the 'Test (Relational) Relational Schema' document, showing options: 'Open ...', 'Rename ...', 'Move to Trash', 'Copy', 'Export', 'Convert to Relational Schema', and 'Generate SQL'. The 'Generate SQL' option is circled in red.

The 'Generate SQL' dialog shows the following SQL code:

```
CREATE TABLE B
(
  Key_B INT NOT NULL,
  PRIMARY KEY (Key_B)
);

CREATE TABLE A
(
  Key_A INT NOT NULL,
  NewAttribute VARCHAR(50) NOT NULL,
  Key_B INT,
  PRIMARY KEY (Key_A),
  FOREIGN KEY (Key_B) REFERENCES B(Key_B)
);

CREATE TABLE C
(
  Partial_key_C INT NOT NULL,
  Key_A INT NOT NULL,
  PRIMARY KEY (Partial_key_C, Key_A),
  FOREIGN KEY (Key_A) REFERENCES A(Key_A)
);
```

Buttons: CLOSE (ESC) COPY

ERD⁺ - Practice (15 minutes)



- Create Entities and Attributes in ERD⁺ for the following:

EMPLOYEE

- Ssn (key, unique)
- Bdate
- Fname
- Minit
- Lname
- Address
- Salary
- Sex

PROJECT

- **Number** (key, unique)
- **Name**
- Location
- Department

DEPENDENT

- Employee
- **Name**
- Relationship
- Bdate
- Sex



EPL342 –Databases

Lab 3

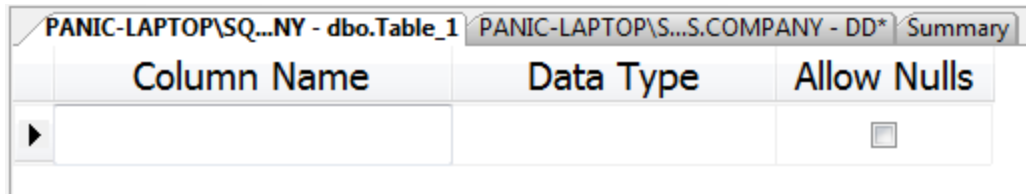
ER Modeling (Entities) in ERD+ Conceptual Modeling in SQL Server 2017



Creating Tables

- First, navigate to your database and right-click on Tables → New Table

The designer provides you with 3 columns:



Column Name	Data Type	Allow Nulls
		<input type="checkbox"/>

Column name: the name of the column (e.g., Name, Birth Date, Salary)

Data type: the data type for the column (e.g., int, varchar(30), bit)

Allow nulls: if checked then you must supply a value for each row (nulls are not allowed)

DEPARTMENT



- Specification I - Each department has the following fields:
 - Name
 - Number
 - Manager
 - Start date of the department manager
 - Multiple locations

Creating Table DEPARTMENT



- You must select the appropriate data type for each column
 - **number:** is an integer (e.g., tinyint, smallint, int, bigint)
 - **name:** is a string (e.g., char, varchar)
 - **Manager:** is the name of an employee (i.e., a string)
 - **Manager start date:** is a date (e.g., datetime, smalldatetime)



Creating Table DEPARTMENT

- Create table department using the following specifications

Column Name	Data Type	Allow Nulls
number	int	No
name	nvarchar(50)	No
Manager	nvarchar(50)	No
Manager start date	Smalldatetime	No

- Save the table with the name DEPARTMENT

Column Names



- Column names can only contain valid characters (i.e., letters, digits and underscores)
- When you saved the DEPARTMENT table note that *Manager start date* is now saved as *[Manager start date]*. This is because white spaces are considered invalid characters.

Column Names Limitations



- To ensure maximum database compatibility
 - **Avoid using special characters**
If you use these characters < [,], ', " > then it is more difficult not only to reference that object but also to read code that contains the name of that object
 - **Avoid using reserved keywords**
SQL Server uses [reserved keywords](#) for manipulating and accessing databases.
 - When you use special characters or reserved keywords the column name is saved using brackets

Column Names/Identifier Rules



- The first character must be one of the following:
 - A letter as defined by the Unicode Standard 3.2. The Unicode definition of letters includes Latin characters from a through z, from A through Z, and also letter characters from other languages.
 - The underscore (`_`), at sign (`@`), or number sign (`#`).
Certain symbols at the beginning of an identifier have special meaning in SQL Server. A regular identifier that starts with the at sign always denotes a local variable or parameter and cannot be used as the name of any other type of object. An identifier that starts with a number sign denotes a temporary table or procedure. An identifier that starts with double number signs (`##`) denotes a global temporary object. Although the number sign or double number sign characters can be used to begin the names of other types of objects, we do not recommend this practice.
Some Transact-SQL functions have names that start with double at signs (`@@`).
- Subsequent characters can include the following:
 - Letters as defined in the Unicode Standard 3.2.
 - Decimal numbers from either Basic Latin or other national scripts.
 - The at sign, dollar sign (`$`), number sign, or underscore.
- The identifier must not be a Transact-SQL reserved word. SQL Server reserves both the uppercase and lowercase versions of reserved words.
- Embedded spaces or special characters are not allowed.
- Supplementary characters are not allowed
- [More info here](#)



Column Properties

Table: Department
Column: number

Column Properties	
(General)	
(Name)	number
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	
Has Non-SQL Server Subscriber	No
Identity Specification	
Indexable	Yes
Merge-published	No
Not For Replication	No
Replicated	No
RowGuid	No
Size	4
(General)	

COMPANY Database



- Create the COMPANY database using the following specifications

DEPENDENT

Column Name	Data Type	Allow Nulls
Relationship	nvarchar(30)	<input type="checkbox"/>
Birth_date	smalldatetime	<input type="checkbox"/>
Sex	bit	<input type="checkbox"/>
Employee	int	<input type="checkbox"/>
Dependent_name	nvarchar(50)	<input type="checkbox"/>

PROJECT

Column Name	Data Type	Allow Nulls
number	int	<input type="checkbox"/>
name	nvarchar(50)	<input type="checkbox"/>
location	nvarchar(50)	<input type="checkbox"/>
controlling_department	nvarchar(50)	<input type="checkbox"/>

EMPLOYEE

Column Name	Data Type	Allow Nulls
SSN	int	<input type="checkbox"/>
Bdate	smalldatetime	<input type="checkbox"/>
Fname	nvarchar(20)	<input type="checkbox"/>
Minit	nvarchar(1)	<input type="checkbox"/>
Lname	nvarchar(30)	<input type="checkbox"/>
Address	nvarchar(100)	<input type="checkbox"/>
Salary	smallmoney	<input type="checkbox"/>
Sex	bit	<input type="checkbox"/>
Department	nvarchar(50)	<input type="checkbox"/>
Supervisor	int	<input type="checkbox"/>





Primary Keys

- A **Primary key** is a candidate key to uniquely identify each row in a table
- Candidate keys:
- DEPARTMENT: name, number
- PROJECT: name, number
- EMPLOYEE: ssn
- DEPENDENT: ???



Creating Primary Keys

- To create a Primary Key, open the table design and select a column.
- Use the  icon to assign the column as the primary key
- Set the following Primary Keys
- DEPARTMENT: number
- PROJECT: number
- EMPLOYEE: ssn

DEPARTMENT	
	number
	name
	Manager
	Manager_start_date

Creating Identities



- Identities are columns with unique values produced automatically from SQL Server
- You can set the following properties:
 - **Is Identity:** Indicates whether or not this column is an identity column
 - **Identity Seed:** The value that will be assigned to the first row in the table
 - **Identity Increment:** This value is the increment that will be added to the **Identity Seed** for each subsequent row

Database Diagram



- Right-click on Database Diagrams and select ***New Database Diagram***
- Use the add button to add all tables to the diagram
- Right-click on a table to see Table View options (e.g., Standard, Keys, Custom, etc)
- Save the diagram as DD
- **In the future we are going to add relationships to the database diagram**

COMPANY database diagram



DEPARTMENT	
<input checked="" type="checkbox"/>	number
<input type="checkbox"/>	name
<input type="checkbox"/>	Manager
<input type="checkbox"/>	Manager_start_date

EMPLOYEE	
<input checked="" type="checkbox"/>	SSN
<input type="checkbox"/>	Bdate
<input type="checkbox"/>	Fname
<input type="checkbox"/>	Minit
<input type="checkbox"/>	Lname
<input type="checkbox"/>	Address
<input type="checkbox"/>	Salary
<input type="checkbox"/>	Sex
<input type="checkbox"/>	Department
<input type="checkbox"/>	Supervisor

DEPENDENT	
<input type="checkbox"/>	Relationship
<input type="checkbox"/>	Birth_date
<input type="checkbox"/>	Sex
<input type="checkbox"/>	Employee
<input type="checkbox"/>	Dependent_name

PROJECT	
<input checked="" type="checkbox"/>	number
<input type="checkbox"/>	name
<input type="checkbox"/>	location
<input type="checkbox"/>	controlling_department



Table Views

Column Names

DEPARTMENT	
	number
	name
	Manager
	Manager_start_date

Standard

DEPARTMENT			
	Column Name	Data Type	Allow Nulls
	number	int	<input type="checkbox"/>
	name	nvarchar(50)	<input type="checkbox"/>
	Manager	nvarchar(50)	<input type="checkbox"/>
	Manager_start_date	smalldatetime	<input type="checkbox"/>

Keys

DEPARTMENT	
	number

Custom

DEPARTMENT						
	Column Name	Nullable	Data Type	Length	Default Value	Description
	number	No	int	4		
	name	No	nvarchar(50)	50		
	Manager	No	nvarchar(50)	50		
	Manager_st...	No	smalldatetime	4		