# Lab 04: ERD Design

## Objective

This lab aims to develop an understanding of ERD designing in Chen's Notation, Crow's-foot notation and using an online tool (DB Designer). Students will be required to set up an account on DB Designer. *Note: Tutorial on how to use the tool has been included towards the end of this document.*

## Submission Requirements:

- This lab is to be completed in teams of **no more than 2 members**.
- Compile your work in a word document (Template is provided on LMS).This will include images of all diagrams along with team member names and ERPs.
- Only one member of the team is required to submit.
- Progress will be recorded by observing teamwork and completion of the task.
- File naming convention to follow: Lab04_Name1(ERP)_Name2(ERP).docx

## Scenario 1: Enrollment

A university registrar's office maintains data about the following entities:

1. **Courses**, including course_code, title, credits, syllabus, and prerequisites.
2. **Course offerings**, including class_number, year, semester, timings, and classroom.
3. **Students**, including studentERP, name, email, D.O.B and program.
4. **Instructors**, including InstructorERP, name, department, and title.

The students are enrolled in courses and are awarded grades.

Task:

1. Draw the ER Model in Chen's notation (on paper). Identify business rules and make assumptions as necessary considering an enrollment system. Write your assumptions in your document. Note: You only need to create the entities as listed above.
2. Convert the ER model to a complete relational model by correctly modelling relationships and represent it on the DBDesigner tool appropriately.
3. Do you think some of these attributes could potentially be entities in the larger database? Pen down your thoughts briefly.

## Scenario 2: Cricket Tournament

A cricket tournament is being held in Pakistan in which 8 different countries are participating. Each country has nominated its 12 players who will participate in the tournament. The system keeps a record of different stadiums of the country, the city they are located in and their capacities. A panel of umpires, belonging to different countries, has been established. For each match, any two umpires will be selected from the panel. A schedule of matches will be published describing the date and time of each match, participating teams, selected umpires, name of the city and the stadium where the match will be held, the winning team and man of the match. The system keeps a record of total runs scored and total wickets taken by each player in every match.

Task:

1. Draw the ER Model in Crow's foot notation (on paper).
2. Draw the complete relational schema (with all FKs mapped) using DBDesigner.

## Scenario 3: Hotel Management System

A hotel management system keeps information about rooms, floors and halls in a hotel and their arrangements. A data model must be designed for this system based on the following information:

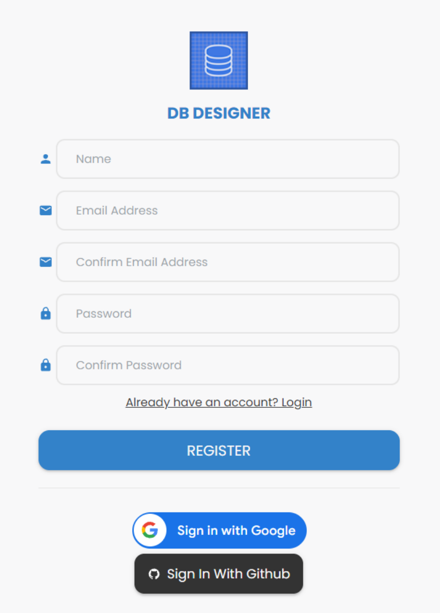
- The hotel has several rooms (RoomNo, Description) located in a multi-floored building and a limited number of dining halls (HallNo, Description) located on different floors.
- Each room is assigned a certain occupancy type (i.e., Single, Double, Twin, Triple, Quad) and a category (e.g. A, B, C etc). The category defines facilities available in that room (e.g., internet access, laundry service, pickup, drop-off etc).
- Each room is assigned a dining hall for breakfast/lunch/dinner.
- The same dining hall can be assigned to multiple rooms and same or different halls can be assigned to a single room for breakfast, lunch and dinner.
- Each floor contains rooms of any one category (e.g. this is not possible to have both category A and B rooms located on the same floor.)
- Room charges are defined on a per night basis for each category and occupancy type. E.g., All rooms belonging to category 'A' and occupancy type 'Double' have same per night charges.

Task:

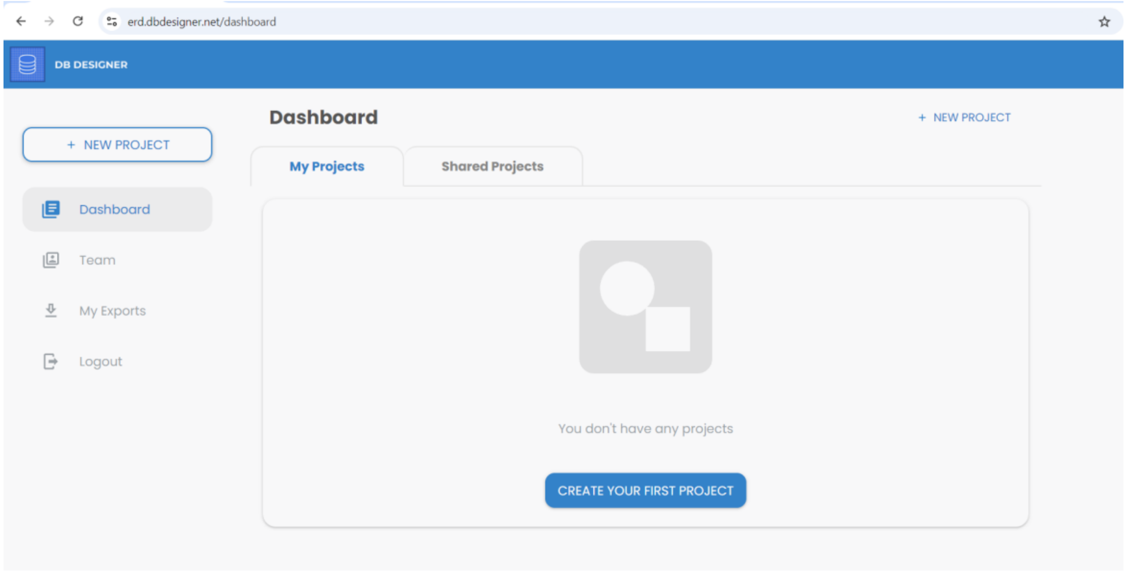
1. Draw the complete relational schema (with all FKs mapped) using DBDesigner.

## DBDesigner Tutorial

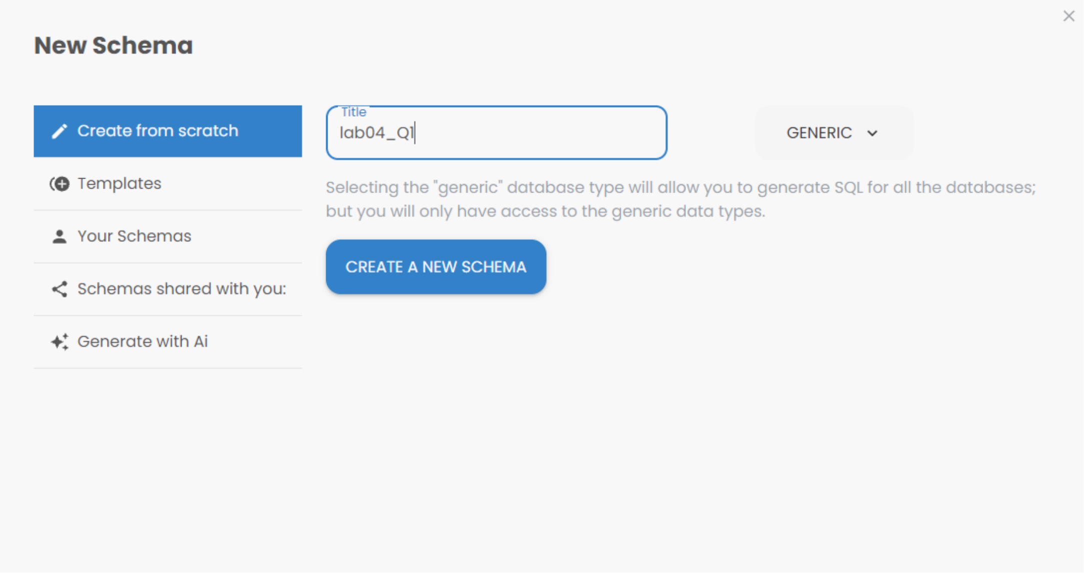
1. Head over to DB Designer: https://erd.dbdesigner.net/register
2. Enter your details and register.



3. You should now have navigated to a Dashboard. Create a new project.



4. Create a new schema from scratch. Give it a suitable name and proceed.

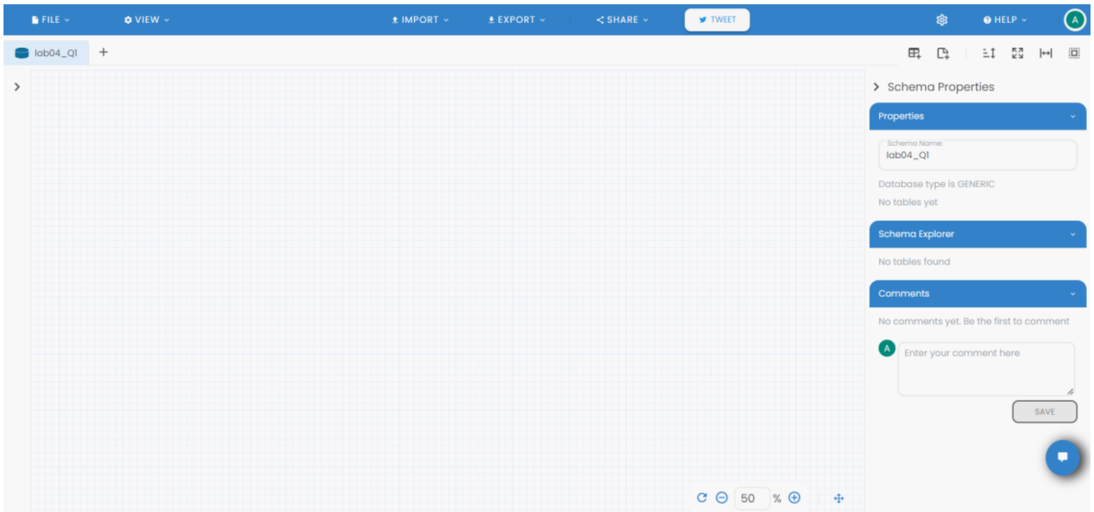


New Schema

Title: lab04_Q1

GENERIC

Selecting the "generic" database type will allow you to generate SQL for all the databases; but you will only have access to the generic data types.



Schema Properties

Schema Name: lab04_Q1

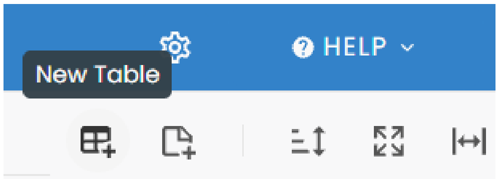
Database type is GENERIC

No tables yet

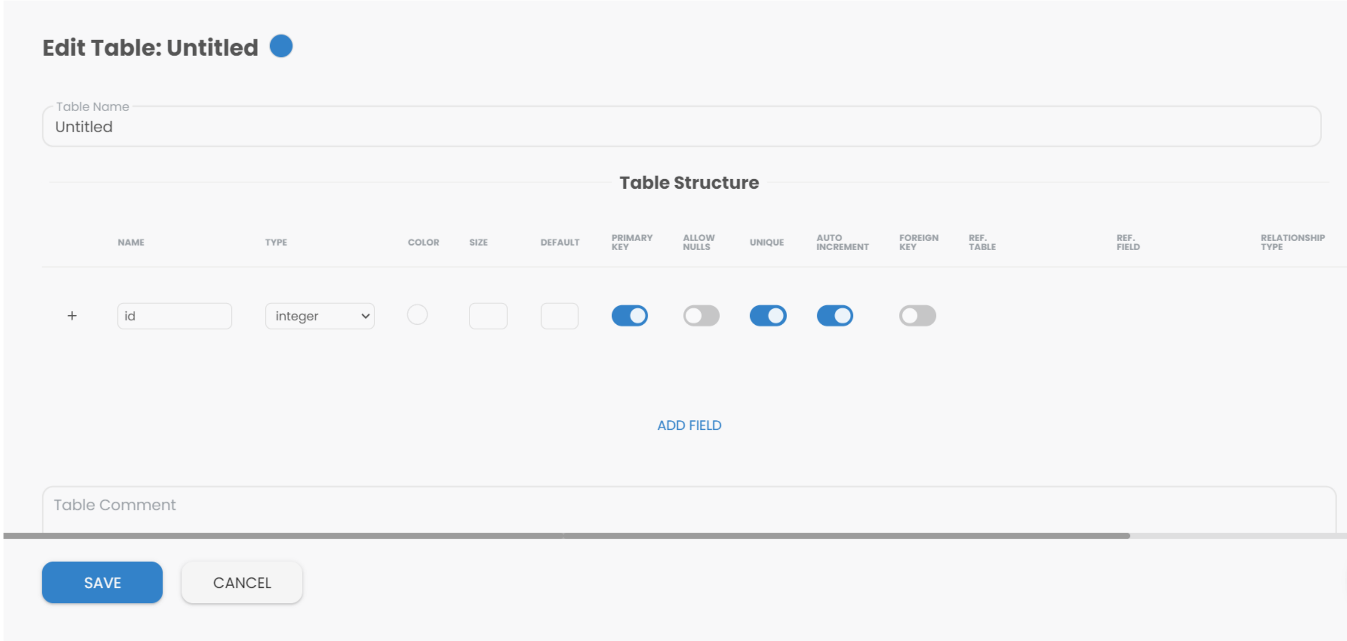
No tables found

No comments yet. Be the first to comment

5. To create a new table, you will find the **New Table** symbol on the top right. Notice, you can also add notes on your diagram using the icon next to it.



6. You will then see a window which will allow you to insert attributes and apply constraints as necessary.



**Edit Table: Untitled**

Table Name: Untitled

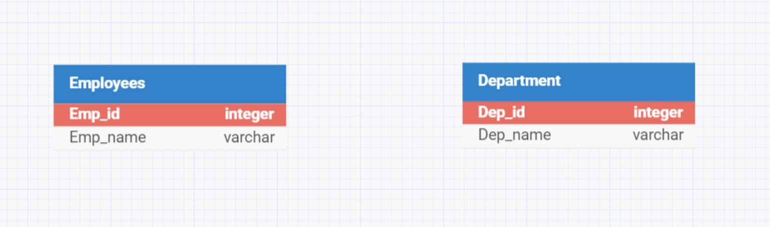
Table Structure

| | NAME | TYPE | COLOR | SIZE | DEFAULT | PRIMARY KEY | ALLOW NULLS | UNIQUE | AUTO INCREMENT | FOREIGN KEY | REF. TABLE | REF. FIELD | RELATIONSHIP TYPE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| + | id | integer | ○ | | | ● | ○ | ● | ● | ○ | | | |

Write the name, select the data type and set the size or default value for the column e.g. *Emp_ID, varchar, 255.* To add constraints to the column e.g. Primary Key, Allow Nulls, Unique, Auto-increment or Foreign Key, toggle the corresponding button.

Note: the datatype options here are slightly different. DBDesigner allows adding data formats compatible to a variety of database types e.g Oracle, mySQL, etc. Once your design is complete, the DBdesigner will allow you to export the design as SQL script which will then change the datatypes as appropriate for e.g varchar will change to varchar2 for Oracle. Later, you also have the option to modify the DDL script should you need to make any further changes.

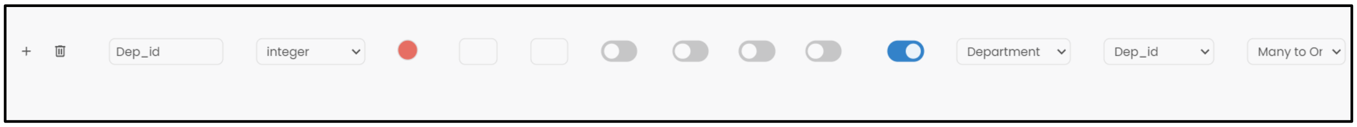
7. Create multiple tables with some attributes.



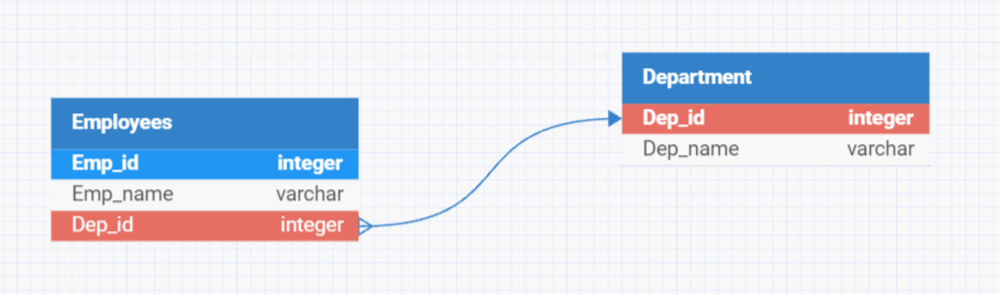
| Employees | |
|---|---|
| **Emp_id** | **integer** |
| Emp_name | varchar |

| Department | |
|---|---|
| **Dep_id** | **integer** |
| Dep_name | varchar |

8. Now we need to establish the relationship between these tables. 1 Employee works in 1..1 Department and 1 Department has 1..M employees working in it. This means we have a One-to-Many Relationship between Department and Employees. In such scenarios, we will move the PK of the one-side of the relationship to the many-side. We need to add Dep_id to the Employees table and apply the Foreign Key constraint on it. Make sure the data types are the same so that there is no conflict.



Note: Here we will enter relationship type as Many-to-One. We need to interpret the relationship type from the table we are making changes in. So many employees can be mapped to a single department. The color selection here is completely optional and is only done to make your diagram look neater.



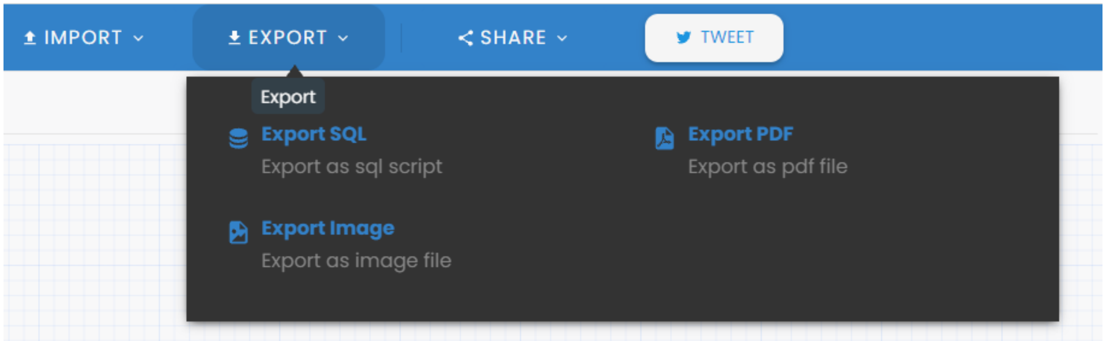
**Employees**

| | |
|---|---|
| Emp_id | integer |
| Emp_name | varchar |
| Dep_id | integer |

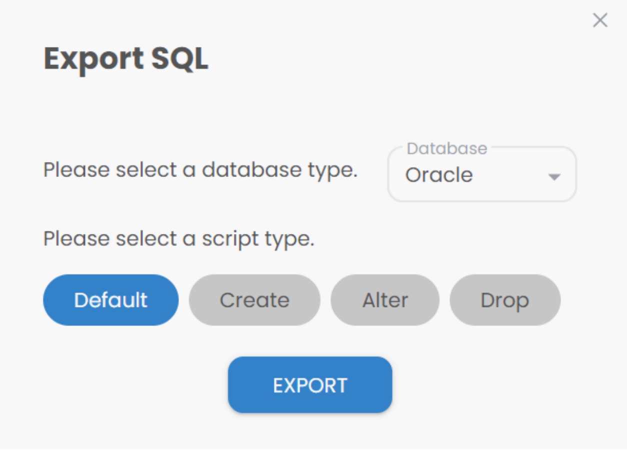
**Department**

| | |
|---|---|
| Dep_id | integer |
| Dep_name | varchar |

9. Once you have completed your design, you should be able to see **Export** on top of your screen**.** Here you have the option to export as SQL, pdf or image file.



10. To export the DDL script for your ERD, Select Export SQL. It allows you to select multiple database types. You may select Oracle here. Your .SQL file will then be downloaded.



**Export SQL**

Please select a database type. Database: Oracle

Please select a script type.