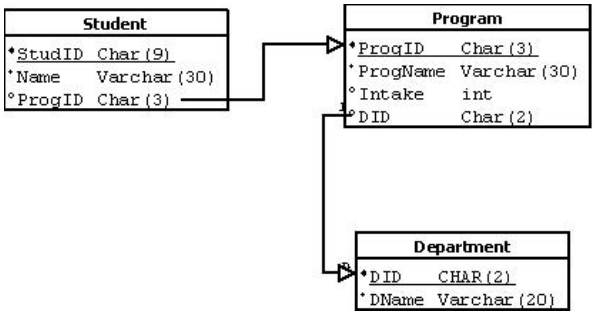
Lab02 Database Schema and SQL-DDL

IT214 Database Management System, Winter'2025; pm_jat @ daiict

In this lab, you attempt to understand relational concepts and learn to define relational schema using SQL-DDL.



(1) Understand the DDL script of Company Schema of Lab01 and create a detailed schema diagram as shown here for XIT schema.

Use the Dia software (http://dia-installer.de/). Though it is desired that you learn Dia and use it for creating schema diagrams. However, if you are not able to use it, you can create a hand sketch and upload its scan. (PS: Dia software allows you to choose a set of graphics primitives that can be used for creating different types of diagrams). For creating a relational schema diagram, you choose the "Database" set of primitives.)

[Deliverable: Schema Diagram]

(2) Understand the DDL script of `DA-Acad` database schema from:
http://intranet.daiict.ac.in/~pm_jat/acad_ddl.sql
Create a similar Schema diagram for this database too (as in question 1)!

[Deliverable: Schema Diagram]

(3) Below is a set of relations for creating a database for the "Sales Management" module of an enterprise application. Here you are given the necessary details for each relation; that is, the list of attributes, its primary key, and foreign key information.

```
customer(cust_id, name, city, state, pin, email)
    -- a tuple of this relation represents a customer
    -- Let cust_id be the Primary Key
items(item_code, item_name, category_id, saleprice, qty_in_stock,
reorderlevel (int), averagepurchaseprice (int) );
    -- a tuple of this relation represents an item
    -- Let item_code be Primary Key
    --  reorderlevel is used to record the required minimum stock required for an item,
       if the stock falls below, it is to be reordered
invoice(invno, invdate, customerid)
    -- a tuple of this relation represents an invoice
    -- Let invno be the Primary Key
    -- customerid is a foreign key referring into the customer table
invoicedetails(invno, itemcode, qty, rate)
    -- used to record details of an invoice; a tuple of this
    -- relation represents an item entry for an invoice
    -- invno is a foreign key referring into the invoice table
    -- itemcode is a foreign key referring into item table
    -- Let {invno, itemcode} jointly be the Primary Key
    -- the attribute rate here indicates the amount per item
```

Note: For all IDs and codes you can use integer types

You are required to Create

(1) the Schema Diagram as in question 1
(2) a DDL script for the schema of this database.

[Deliverable: Schema Diagram, and DDL script]

Submission Instructions: Submit everything in a single PDF file.