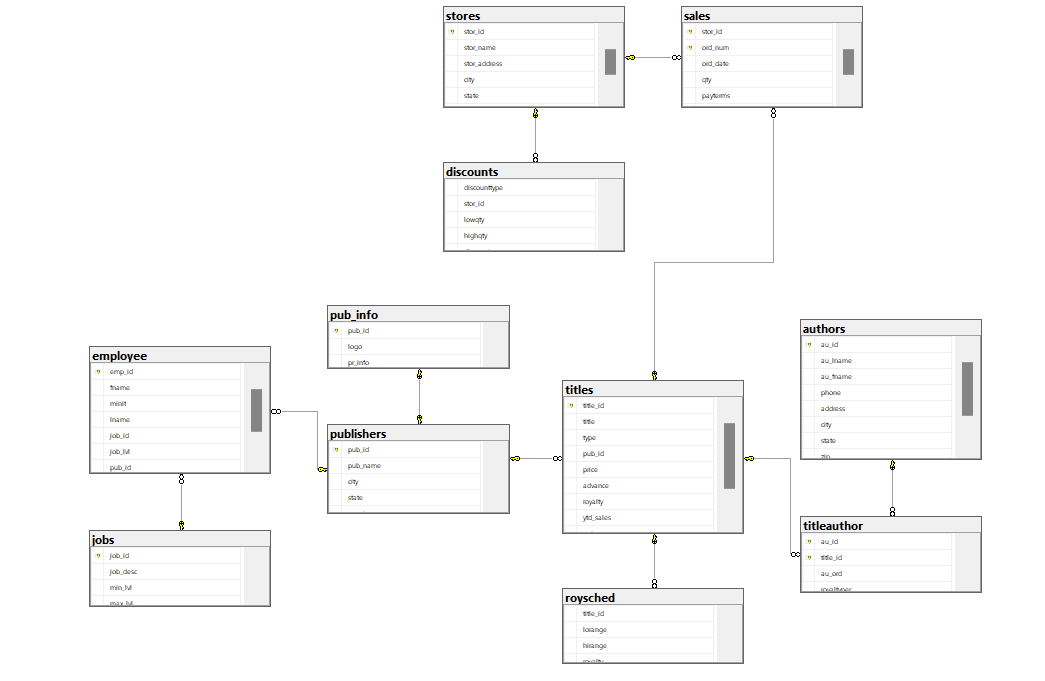
**Day 2 – DBMS – SQL Queries**



1) Print all the titles names

2) Print all the titles that have been published by 1389

3) Print the books that have price in range of 10 to 15

4) Print those books that have no price

5) Print the book names that starts with 'The'

6) Print the book names that do not have 'v' in their name

7) print the books sorted by the royalty

8) print the books sorted by publisher in descending then by types in ascending then by price in descending

9) Print the average price of books in every type

10) print all the types in unique

11) Print the first 2 costliest books

12) Print books that are of type business and have price less than 20 which also have advance greater than 7000

13) Select those publisher id and number of books which have price between 15 to 25 and have 'It' in its name. Print only those which have count greater than 2. Also sort the result in ascending order of count

14) Print the Authors who are from 'CA'

15) Print the count of authors from every state

1. select title from titles;  
     
   A screenshot of a computer

   AI-generated content may be incorrect.
2. select \* from titles where pub\_id = 1389;

A screenshot of a computer

AI-generated content may be incorrect.

1. select \* from titles where price >= 10 and price <= 15;

A screenshot of a computer

AI-generated content may be incorrect.

1. select \* from titles where price IS NULL;

A screenshot of a computer

AI-generated content may be incorrect.

1. select \* from titles where title like 'The%';

A screenshot of a computer

AI-generated content may be incorrect.

1. select \* from titles where title not like '%v%';

A screenshot of a computer

AI-generated content may be incorrect.

1. select \* from titles order by royalty;

A screenshot of a computer

AI-generated content may be incorrect.

1. select \* from titles t, publishers p where t.pub\_id = p.pub\_id order by p.pub\_name desc, t.type asc, t.price desc;

A screenshot of a computer

AI-generated content may be incorrect.

1. select type, avg(price) as average\_price from titles group by type;

A screenshot of a computer

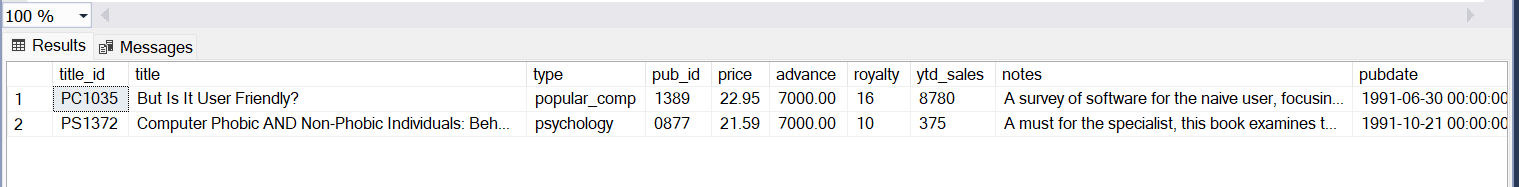
AI-generated content may be incorrect.

1. select distinct type from titles;

A screenshot of a computer

AI-generated content may be incorrect.

1. select top 2 \* from titles order by price desc;



1. select \* from titles where type = 'business' and price < 20 and advance > 7000;

A screenshot of a computer

AI-generated content may be incorrect.

1. SELECT pub\_id, count(\*) as number\_books from titles

where price between 15 and 25 and title like '%it%'

group by pub\_id having count(\*) > 2

order by number\_books;

A screenshot of a computer

AI-generated content may be incorrect.

1. SELECT \* from authors where state = 'CA';

A screenshot of a computer

AI-generated content may be incorrect.

1. SELECT state,count(\*) as count from authors group by state;

A screenshot of a computer

AI-generated content may be incorrect.

Design the database for a shop which sells products  
Points for consideration  
  1) One product can be supplied by many suppliers  
  2) One supplier can supply many products  
  3) All customers details have to present  
  4) A customer can buy more than one product in every purchase  
  5) Bill for every purchase has to be stored  
  6) These are just details of one shop

categories  
id, name, status

country  
id, name

state  
id, name, country\_id

City  
id, name, state\_id

area  
zipcode, name, city\_id

address  
id, door\_number, addressline1, zipcode

supplier  
id, name, contact\_person, phone, email, address\_id, status

product  
id, Name, unit\_price, quantity, description, image

product\_supplier  
transaction\_id, product\_id, supplier\_id, date\_of\_supply, quantity,

Customer  
id, Name, Phone, age, address\_id

order  
  order\_number, customer\_id, Date\_of\_order, amount, order\_status

order\_details  
  id, order\_number, product\_id, quantity, unit\_price

SQL Query:

CREATE TABLE Categories (

id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

status VARCHAR(50) NOT NULL

);

CREATE TABLE Country (

id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL

);

CREATE TABLE State (

id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

country\_id INT NOT NULL,

FOREIGN KEY (country\_id) REFERENCES Country(id)

);

CREATE TABLE City (

id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

state\_id INT NOT NULL,

FOREIGN KEY (state\_id) REFERENCES State(id)

);

CREATE TABLE Area (

zipcode VARCHAR(20) PRIMARY KEY,

name VARCHAR(255) NOT NULL,

city\_id INT NOT NULL,

FOREIGN KEY (city\_id) REFERENCES City(id)

);

CREATE TABLE Address (

id INT PRIMARY KEY,

door\_number VARCHAR(50),

addressline1 TEXT NOT NULL,

zipcode VARCHAR(20) NOT NULL,

FOREIGN KEY (zipcode) REFERENCES Area(zipcode)

);

CREATE TABLE Supplier (

id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

contact\_person VARCHAR(255),

phone VARCHAR(20) UNIQUE NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

address\_id INT NOT NULL,

status VARCHAR(50) NOT NULL,

FOREIGN KEY (address\_id) REFERENCES Address(id)

);

CREATE TABLE Product (

id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

unit\_price DECIMAL(10,2) NOT NULL,

quantity INT NOT NULL,

description TEXT,

image VARCHAR(255) -- Store image URL or file path

);

CREATE TABLE ProductSupplier (

transaction\_id INT PRIMARY KEY,

product\_id INT NOT NULL,

supplier\_id INT NOT NULL,

date\_of\_supply DATE NOT NULL,

quantity INT NOT NULL,

FOREIGN KEY (product\_id) REFERENCES Product(id),

FOREIGN KEY (supplier\_id) REFERENCES Supplier(id)

);

CREATE TABLE Customer (

id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

phone VARCHAR(20) UNIQUE NOT NULL,

age INT CHECK (age > 0),

address\_id INT NOT NULL,

FOREIGN KEY (address\_id) REFERENCES Address(id)

);

CREATE TABLE Orders (

order\_number INT PRIMARY KEY,

customer\_id INT NOT NULL,

date\_of\_order DATETIME DEFAULT CURRENT\_TIMESTAMP,

amount DECIMAL(10,2) NOT NULL,

order\_status VARCHAR(50) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES Customer(id)

);

CREATE TABLE OrderDetails (

id INT PRIMARY KEY,

order\_number INT NOT NULL,

product\_id INT NOT NULL,

quantity INT NOT NULL CHECK (quantity > 0),

unit\_price DECIMAL(10,2) NOT NULL,

FOREIGN KEY (order\_number) REFERENCES Orders(order\_number),

FOREIGN KEY (product\_id) REFERENCES Product(id)

);

Afternoon:

Table Schema:

Create Tables with Integrity Constrains:

a) EMP (empno - primary key, empname, salary, deptname - references entries in a deptname of department table with null constraint, bossno - references entries in an empno of emp table with null constraint)

b) DEPARTMENT (deptname - primary key, floor, phone, empno - references entries in an empno of emp table not null)

c) SALES (salesno - primary key, saleqty, itemname -references entries in a itemname of item table with not null constraint, deptname - references entries in a deptname of department table with not null constraint)

d) ITEM (itemname - primary key, itemtype, itemcolor)

create table item (

itemname varchar(255) primary key,

itemtype varchar(100) not null,

itemcolor varchar(100) not null

);

create table emp (

empno int primary key,

empname varchar(255) not null,

salary decimal(10,2) check(salary > 0),

deptname varchar(255) null,

bossno int null,

foreign key (deptname) references department(deptname),

foreign key (bossno) references emp(empno)

);

create table department (

deptname varchar(255) primary key,

floor int not null check(floor > 0),

phone varchar(20) unique not null,

mgrid int not null,

foreign key (mgrid) references emp(empno)

);

create table sales (

salesno int primary key,

saleqty int not null check(saleqty > 0),

itemname varchar(255) not null,

deptname varchar(255) not null,

foreign key (itemname) references item(itemname),

foreign key (deptname) references department(deptname)

);

ALTER TABLE department

ALTER COLUMN mgrid INT NULL;

INSERT INTO department (deptname, floor, phone, mgrid) VALUES

('Management', 5, 34, NULL);

INSERT INTO department (deptname, floor, phone) VALUES

('Marketing', 2, '1234567891'),

('Accounting', 3, '1234567892'),

('Purchasing', 4, '1234567893'),

('Personnel', 5, '1234567894'),

('Navigation', 6, '1234567895'),

('Books', 7, '1234567896'),

('Clothes', 8, '1234567897'),

('Equipment', 9, '1234567898'),

('Furniture', 10, '1234567899'),

('Recreation', 11, '1234567800');

INSERT INTO emp (empno, empname, salary, deptname, bossno) VALUES  
(1, 'Alice', 75000, 'Management', NULL),  
(2, 'Ned', 45000, 'Marketing', 1),  
(3, 'Andrew', 25000, 'Marketing', 2),  
(4, 'Clare', 22000, 'Marketing', 2),  
(5, 'Todd', 38000, 'Accounting', 1),  
(6, 'Nancy', 22000, 'Accounting', 5),  
(7, 'Brier', 43000, 'Purchasing', 1),  
(8, 'Sarah', 56000, 'Purchasing', 7),  
(9, 'Sophile', 35000, 'Personnel', 1),  
(10, 'Sanjay', 15000, 'Navigation', 3),  
(11, 'Rita', 15000, 'Books', 4),  
(12, 'Gigi', 16000, 'Clothes', 4),  
(13, 'Maggie', 11000, 'Clothes', 4),  
(14, 'Paul', 15000, 'Equipment', 3),  
(15, 'James', 15000, 'Equipment', 3),  
(16, 'Pat', 15000, 'Furniture', 3),  
(17, 'Mark', 15000, 'Recreation', 3);

ALTER TABLE item

ALTER COLUMN itemcolor VARCHAR(100) NULL;

INSERT INTO item (itemname, itemtype, itemcolor) VALUES

('Pocket Knife-Nile', 'E', 'Brown'),

('Pocket Knife-Avon', 'E', 'Brown'),

('Compass', 'N', NULL),

('Geo positioning system', 'N', NULL),

('Elephant Polo stick', 'R', 'Bamboo'),

('Camel Saddle', 'R', 'Brown'),

('Sextant', 'N', NULL),

('Map Measure', 'N', NULL),

('Boots-snake proof', 'C', 'Green'),

('Pith Helmet', 'C', 'Khaki'),

('Hat-polar Explorer', 'C', 'White'),

('Exploring in 10 Easy Lessons', 'B', NULL),

('Hammock', 'F', 'Khaki'),

('How to win Foreign Friends', 'B', NULL),

('Map case', 'E', 'Brown'),

('Safari Chair', 'F', 'Khaki'),

('Safari cooking kit', 'F', 'Khaki'),

('Stetson', 'C', 'Black'),

('Tent - 2 person', 'F', 'Khaki'),

('Tent - 8 person', 'F', 'Khaki');

INSERT INTO sales (salesno, saleqty, itemname, deptname) VALUES

(101, 2, 'Boots-snake proof', 'Clothes'),

(102, 1, 'Pith Helmet', 'Clothes'),

(103, 1, 'Sextant', 'Navigation'),

(104, 3, 'Hat-polar Explorer', 'Clothes'),

(105, 5, 'Pith Helmet', 'Equipment'),

(106, 2, 'Pocket Knife-Nile', 'Clothes'),

(107, 3, 'Pocket Knife-Nile', 'Recreation'),

(108, 1, 'Compass', 'Navigation'),

(109, 2, 'Geo positioning system', 'Navigation'),

(110, 5, 'Map Measure', 'Navigation'),

(111, 1, 'Geo positioning system', 'Books'),

(112, 1, 'Sextant', 'Books'),

(113, 3, 'Pocket Knife-Nile', 'Books'),

(114, 1, 'Pocket Knife-Nile', 'Navigation'),

(115, 1, 'Pocket Knife-Nile', 'Equipment'),

(116, 1, 'Sextant', 'Clothes'),

(117, 1, 'Sextant', 'Equipment'),

(118, 1, 'Sextant', 'Recreation'),

(119, 1, 'Sextant', 'Furniture'),

(120, 1, 'Pocket Knife-Nile', 'Furniture'),

(121, 1, 'Exploring in 10 easy lessons', 'Books'),

(122, 1, 'How to win foreign friends', 'Books'),

(123, 1, 'Compass', 'Books'),

(124, 1, 'Pith Helmet', 'Books'),

(125, 1, 'Elephant Polo stick', 'Recreation'),

(126, 1, 'Camel Saddle', 'Recreation');

UPDATE department SET mgrid = 1 WHERE deptname = 'Management';

UPDATE department SET mgrid = 4 WHERE deptname = 'Books';

UPDATE department SET mgrid = 4 WHERE deptname = 'Clothes';

UPDATE department SET mgrid = 3 WHERE deptname = 'Equipment';

UPDATE department SET mgrid = 3 WHERE deptname = 'Furniture';

UPDATE department SET mgrid = 3 WHERE deptname = 'Navigation';

UPDATE department SET mgrid = 4 WHERE deptname = 'Recreation';

UPDATE department SET mgrid = 5 WHERE deptname = 'Accounting';

UPDATE department SET mgrid = 7 WHERE deptname = 'Purchasing';

UPDATE department SET mgrid = 9 WHERE deptname = 'Personnel';

UPDATE department SET mgrid = 2 WHERE deptname = 'Marketing';

select \* from emp;

A screenshot of a computer

AI-generated content may be incorrect.

select \* from item;

A screenshot of a computer

AI-generated content may be incorrect.

select \* from department;

A screenshot of a computer

AI-generated content may be incorrect.

select \* from sales;  
A screenshot of a computer

AI-generated content may be incorrect.