PROQMAN – PROCUREMENT MANAGEMENT TOOL FOR BUSINESSES

A Web based solution for Enterprises and Business to Manage their Direct and Indirect Procurements

Database Design Document.

Braj Kishor
2019HT13096@wilp.bits-pilani.ac.in
As Part of DBMS Assignment 1

Table of Contents

Context:

This document details out Detailed Database design of the ProQMan solution. It contains the Details of Various Entities, ER Diagram, Relational Design and other details for the solution.

About ProQMan (Features):

ProQMan is a Web based solution for businesses and enterprises to allow them to manage their Direct and Indirect procurements.

The solution allows them to store, track and audit their procurement requests & orders, manage contracts with suppliers and provides pan organisation visibility in the procurement practices.

It also allows the cross sharing and collaboration of contracts and procurements across various sub units and division in enterprises.

The application additionally allows detailed reporting related to procurement expenses and supplier performances allowing companies to mitigate losses related to procurements.

The solution is web based and can be deployed on premise or in cloud and can be made available on the organisation's Intranet.

What is Direct and Indirect procurement?

Direct procurement involves the purchasing of goods, materials, and services directly associated with the production of goods and services that a company is providing. Whereas, indirect spend refers to expenses incurred for materials, services, and maintenance required to operate the business. Both are important for a business, and one can't exist without the other.

Important Terminology

1. Contract: A deal between supplier and the organisation detailing out product and pricing details.

2. Purchase Requisition (PR): A purchase requisition is a form that an internal department of your company submits to the company's purchasing department listing items it wants the purchasing department to order from an outside supplier.
3. Purchase Order (PO): Also known as a PO, the purchase order is a document outlining the details of an actual purchase. It contains details of product, quantity, delivery details etc and is sent to the external supplier.
4. Goods Received (GR): A GR document is generated once the product is received by the company. It contains details of delivery, quantity received, defects etc.

ProQMan Application Solution details

The solution is deployed as web application on the internal network of the organisation. It is a web-based solution with a central database system. The solution can be deployed on-premise or can be offered on cloud such as AWS as SaaS.

As the solution can contain sensitive and resource critical data, the solution uses Database Replication and encryption techniques. The database uses primarily an enterprise grade RDBMS system. Additional tools to ensure high availability and data caching is also used, which is not detailed in this document.

Users of the system

The system is designed to be used for procurement management and auditing. Following are the primary users of the system.

1. CPO (Chief Procurement Officer)
2. Procurement Managers
3. Procurement Executives
4. Category Managers
5. Category Analyst
6. Category Executive
7. CFO (Chief Finance Officer)
8. Finance Managers
9. Finance Auditors

In addition, the **System Administrators** also available whose role is support and maintenance of the system.

User Load

The main user base are the procurement and finance officers and analysts of the company. The solution is designed to be Single Instance per Organisation type solution. Depending on the size of the organisation, the number of users can vary from 50-1000 users and more.

Furthermore, the solution expects hundreds of concurrent users of the system.

Benefits of the Application

The key benefits are

1. Works as a central repository for supplier contracts across Organisation.
2. Allows easy tracking and auditing of procurement requests online.
3. Provides Visibility in procurement practices.
4. Allows dynamic reporting of Procurement expenses and supplier performances.
5. Allows the cross sharing and collaboration of contracts and procurements across various sub units and division in enterprises
6. Allows easy creation of Purchase Requisitions for any unit's category analysts and manager.
7. Allows easy approvals of PR and creation of POs
8. Allows tracking of Goods Received, Online and Hard Copy Receipts, Delivery and other details etc for auditing.
9. Ease in Auditing for Finance teams.

Entities in the system

Described below are some of the Basic Entities of the system in the simplest form of the product. However, the product can have additional entities and relationship as the complexity of the requirements increases.

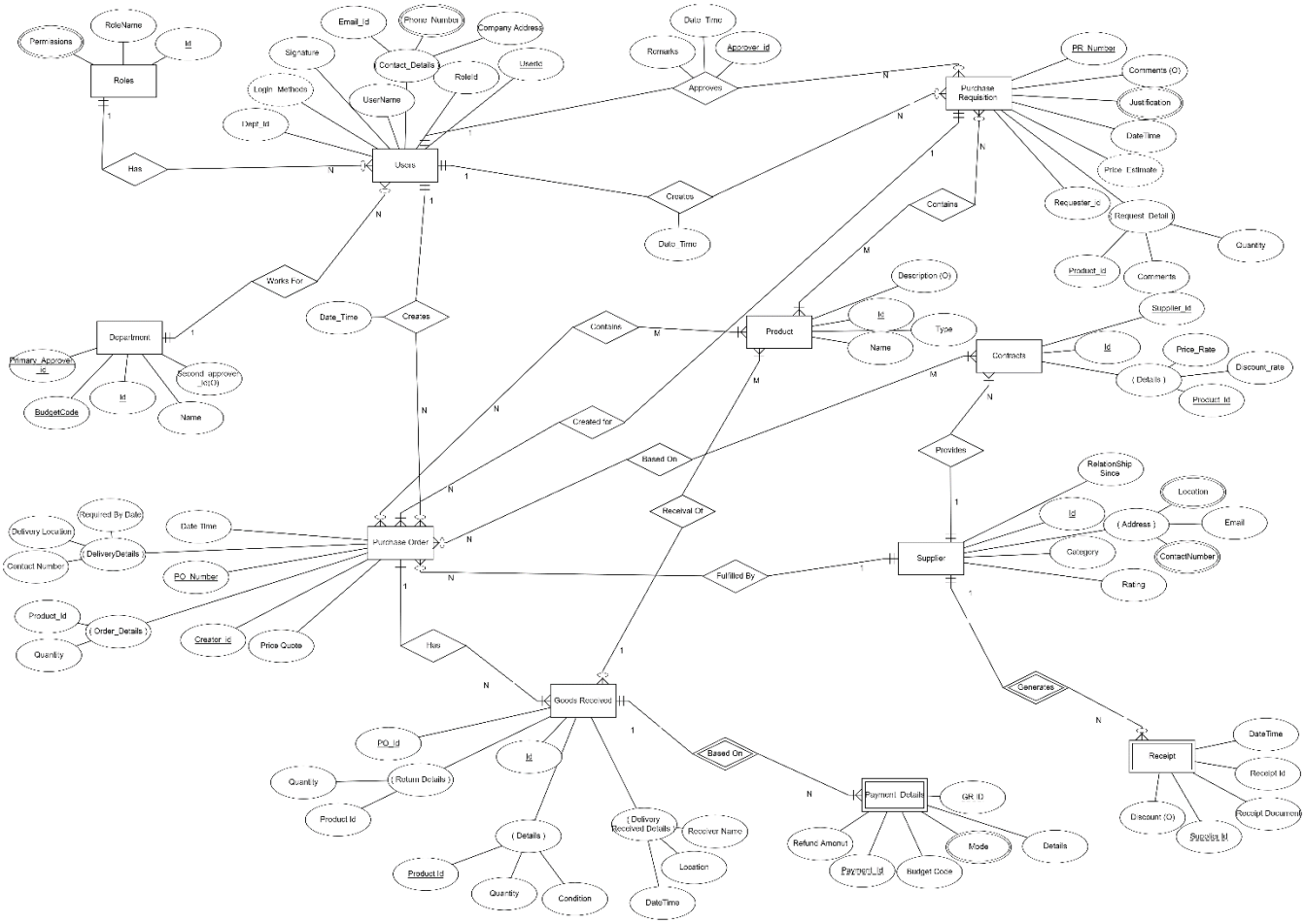
1. Users: There are various types of users of the system. Each can have various types of roles and can be associated with different departments.
2. Roles: The various roles of the users in the company related to the solution.
3. Departments: The departments of the company. Different department has different category for consumables and hence different Category Analysts and Managers
4. Products: These contains the direct and indirect consumables of the organisation.
5. Contracts: The contract contains the pricing deals for various products between organisation and suppliers. It's binding, however, may change upon consent as required.
6. Supplier: The vendor organisation which provides the product for procurement to the organisation. The supplier can be manufacturer, the whole seller or third party.
7. Purchase Requisition: The PR which is created by various departments for acquisition of the products. One PR may contain details of multiple products. It contains quantity required, purpose of the usage etc and is sent for approval.
8. Purchase Order: The PO is an approved PR which is sent outside to the supplier containing details of purchase request like order quantity, contract-based estimate, delivery estimates etc
9. Payment Details: The actual details of the payment made. This includes cost, payment made, payment mode and details like budget code etc. It also contains any prepayment details and refund details.
10. Receipt: The receipt document sent by the supplier. The document contains the total cost, discount, product, quantity, price and any payment details as per supplier. The details can be sent in hard copy or soft copy. The document is directly stored into the database after verification.
11. Goods Received: The details of the goods received by the organisation. This contains the products received along with quantity, Quality of the goods received, Goods returned, delivery details etc.

ER Diagram of the system

Below is the preview of the ER Diagram.

Please use this embedded high-quality image to zoom and see and the details.

ERDiagram.png

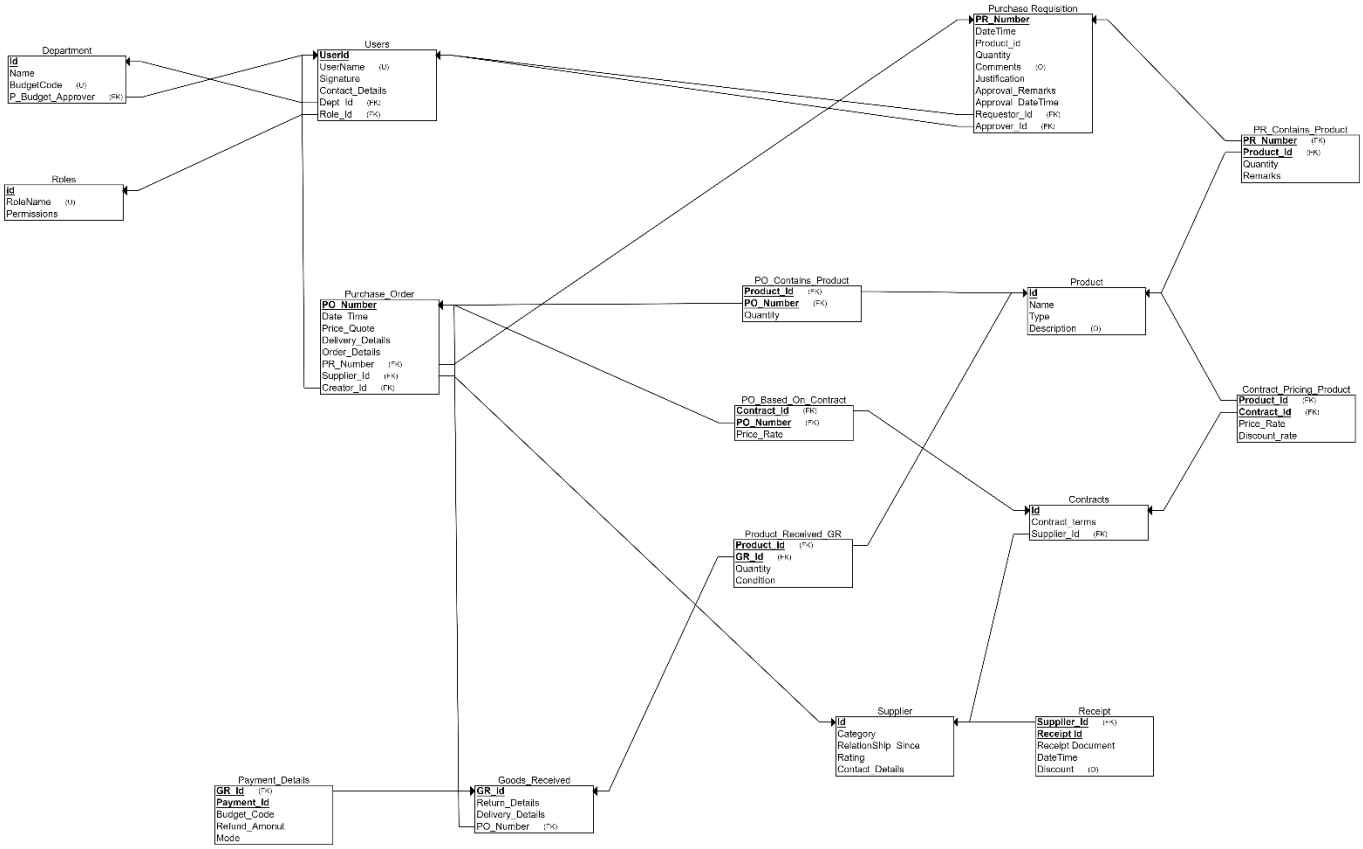


Relational Model

Below is the preview of the Relational Model.

Please use this embedded high-quality image to zoom and see and the details.

RelationalModel.png



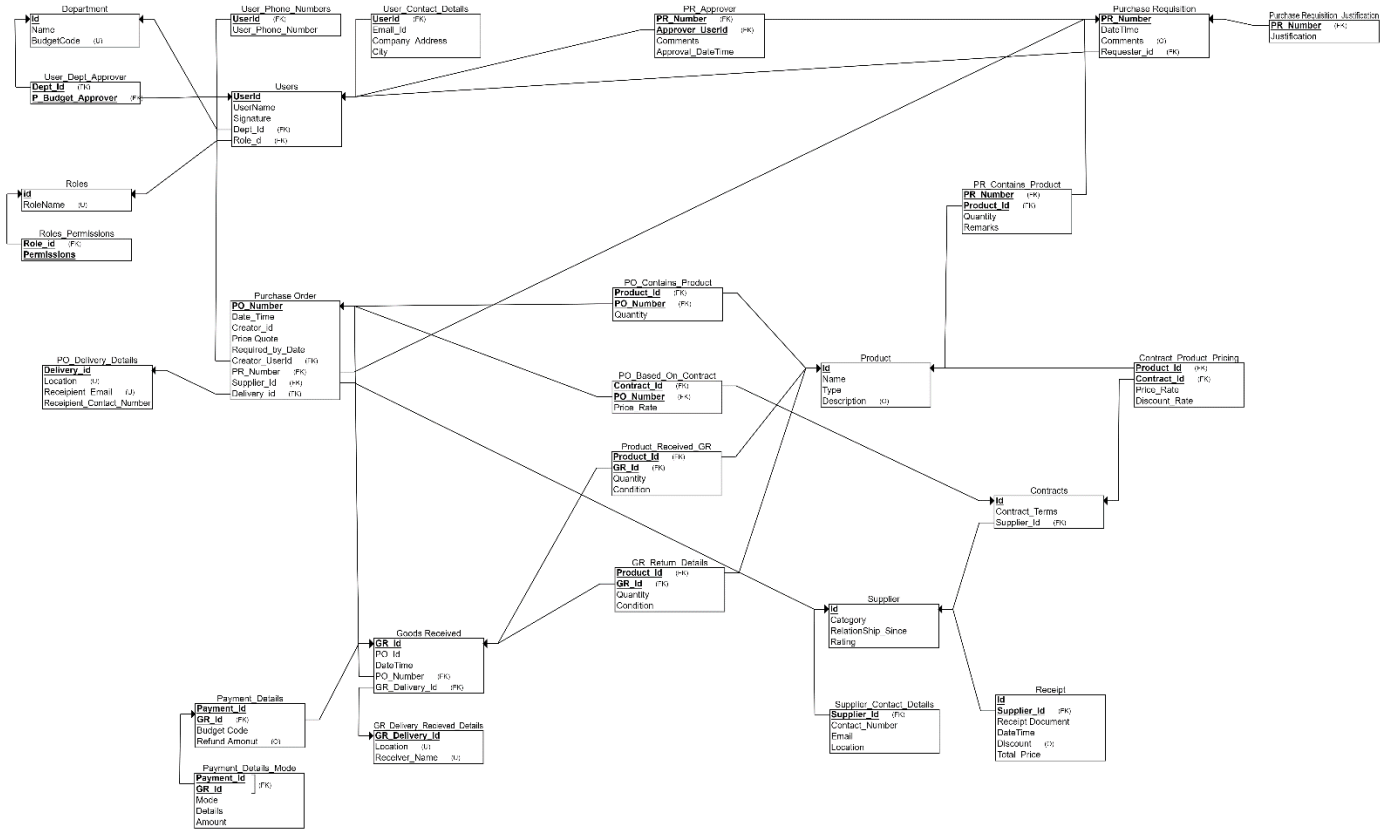
Normalised Relational Model (BCNF)

Below is the preview of the Final Relational Model in BCNF.

Please use this embedded high-quality image to zoom and see and the details.

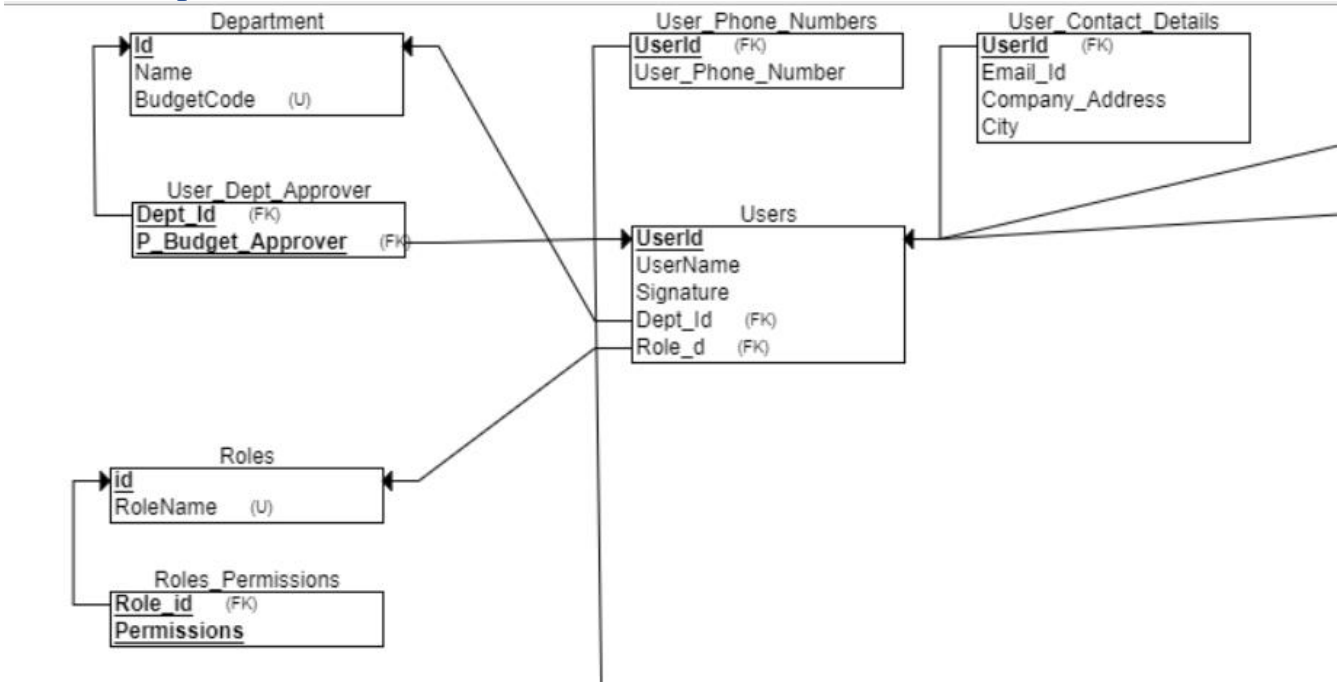
NormalisedRelationalModel.png

Also, PFB the entity based Normalised Diagrams. The final image is aggregation of the all the entity-based images.

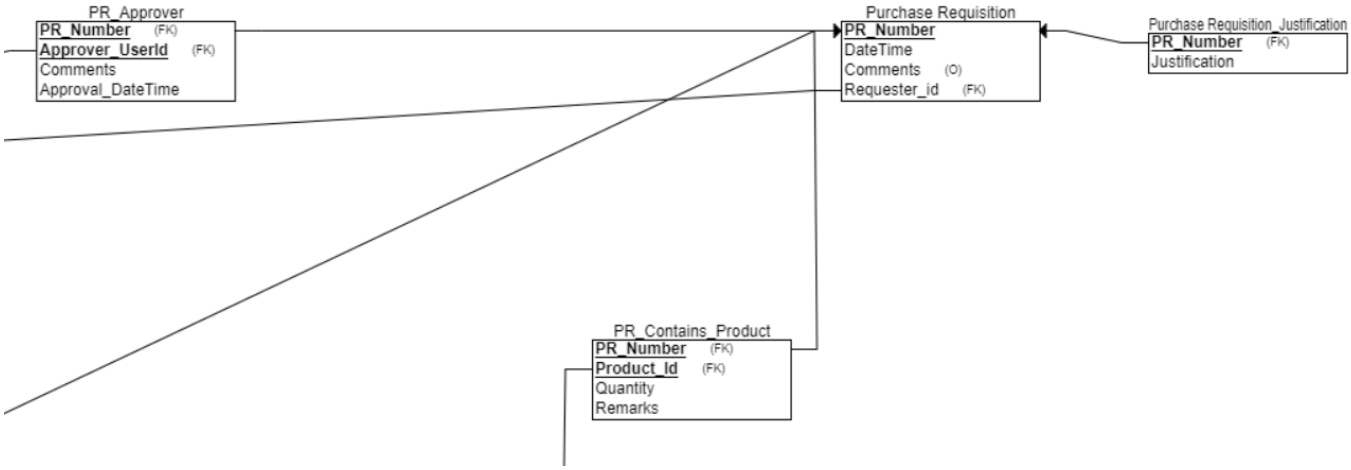


Entity-Based Normalised Diagrams

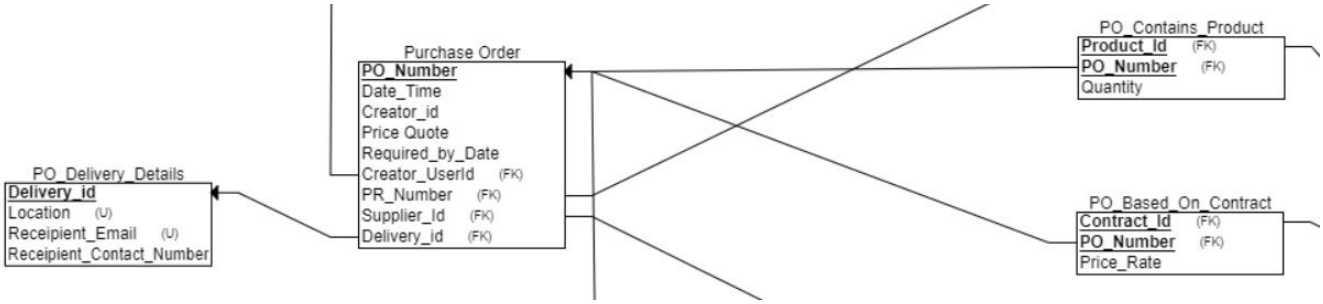
User – Department – Role Entities



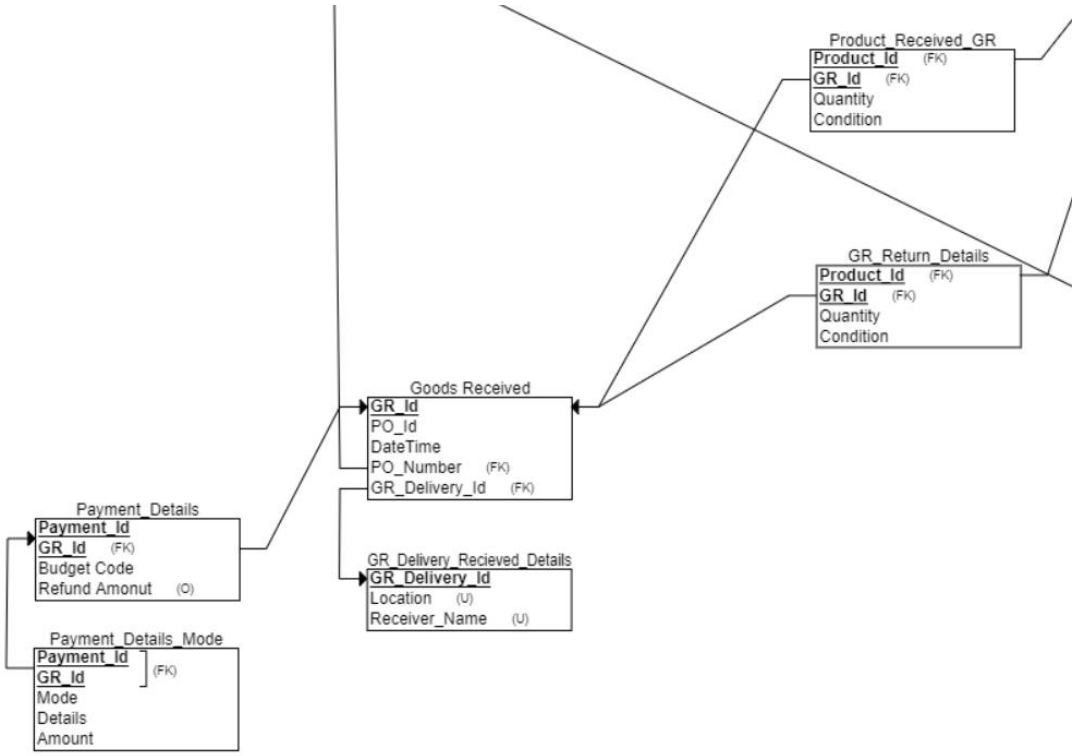
Purchase Requisition Entity



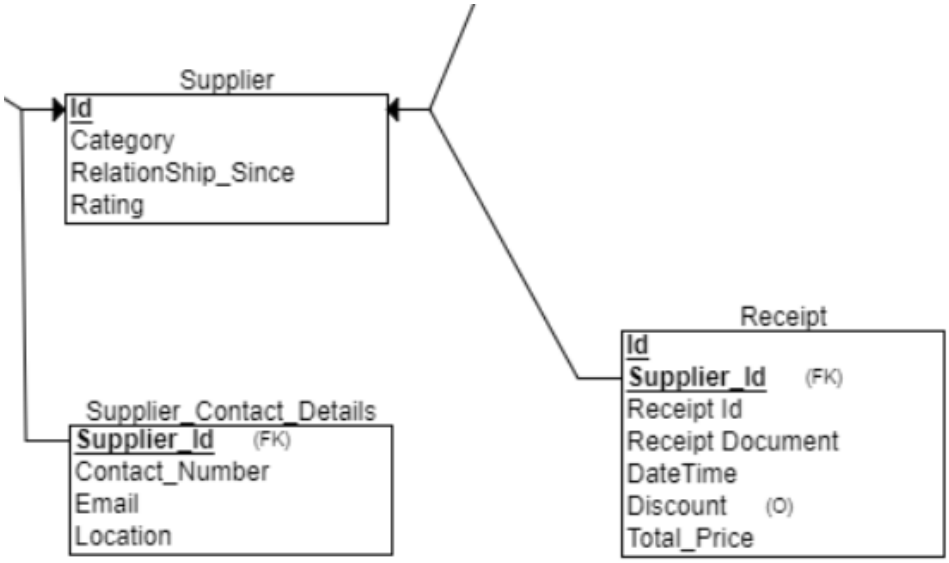
Purchase Order Entity



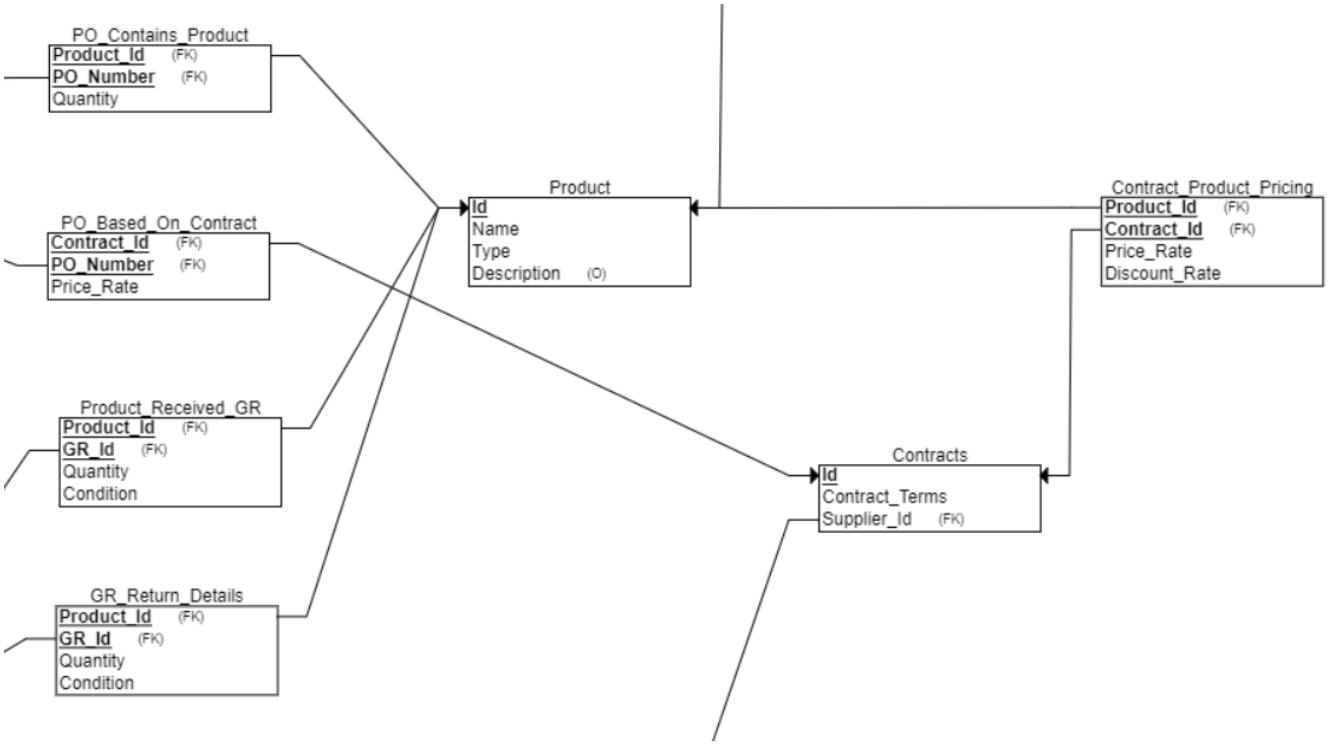
Goods Received – Payment Details



Supplier – Receipt Entity



Contract – Product Entity



SQL Queries

Create SQL:

```
CREATE TABLE Department
(
  Id INT NOT NULL,
  Name VARCHAR(50) NOT NULL,
  BudgetCode VARCHAR NOT NULL,
  PRIMARY KEY (Id),
  UNIQUE (BudgetCode)
);

CREATE TABLE Roles
(
  RoleName ENUM NOT NULL,
  id INT NOT NULL,
  PRIMARY KEY (id),
  UNIQUE (RoleName)
);

CREATE TABLE Product
(
  Id INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  Type ENUM NOT NULL,
  Description Text,
  PRIMARY KEY (Id)
);

CREATE TABLE Supplier
(
  Id INT NOT NULL,
  Category ENUM NOT NULL,
  RelationShip_Since DATE NOT NULL,
  Rating INT NOT NULL,
  PRIMARY KEY (Id)
);

CREATE TABLE Receipt
(
  Id INT NOT NULL,
  Receipt_Document Blob NOT NULL,
  DateTime DATE NOT NULL,
  Discount NUMERIC,
  Total_Price NUMERIC NOT NULL,
  Supplier_Id INT NOT NULL,
  PRIMARY KEY (Id, Supplier_Id),
  FOREIGN KEY (Supplier_Id) REFERENCES Supplier(Id)
);
```

```
CREATE TABLE Roles_Permissions
(
  Permissions INT NOT NULL,
  Role_id INT NOT NULL,
  PRIMARY KEY (Permissions, Role_id),
  FOREIGN KEY (Role_id) REFERENCES Roles(id)
);

CREATE TABLE Supplier_Contact_Details
(
  Contact_Number INT NOT NULL,
  Email VARCHAR(50) NOT NULL,
  Location VARCHAR(100) NOT NULL,
  Supplier_Id INT NOT NULL,
  PRIMARY KEY (Supplier_Id),
  FOREIGN KEY (Supplier_Id) REFERENCES Supplier(Id)
);

CREATE TABLE PO_Delivery_Details
(
  Receipient_Contact_Number VARCHAR(15) NOT NULL,
  Location VARCHAR(50) NOT NULL,
  Receipient_Email VARCHAR(50) NOT NULL,
  Delivery_id INT NOT NULL,
  PRIMARY KEY (Delivery_id),
  UNIQUE (Location),
  UNIQUE (Receipient_Email)
);

CREATE TABLE GR_Delivery_Recieved_Details
(
  Location INT NOT NULL,
  Receiver_Name INT NOT NULL,
  GR_Delivery_Id INT NOT NULL,
  PRIMARY KEY (GR_Delivery_Id),
  UNIQUE (Location),
  UNIQUE (Receiver_Name)
);

CREATE TABLE Users
(
  UserName INT NOT NULL,
  UserId INT NOT NULL,
  Signature Blob NOT NULL,
  Dept_Id INT NOT NULL,
  Role_d INT NOT NULL,
  PRIMARY KEY (UserId),
  FOREIGN KEY (Dept_Id) REFERENCES Department(Id),
  FOREIGN KEY (Role_d) REFERENCES Roles(id)
);
```

```
CREATE TABLE Contracts
(
  Id INT NOT NULL,
  Contract_Terms Text NOT NULL,
  Supplier_Id INT NOT NULL,
  PRIMARY KEY (Id),
  FOREIGN KEY (Supplier_Id) REFERENCES Supplier(Id)
);

CREATE TABLE Purchase_Requisition
(
  PR_Number INT NOT NULL,
  DateTime DATE NOT NULL,
  Comments Text,
  Requester_id INT NOT NULL,
  PRIMARY KEY (PR_Number),
  FOREIGN KEY (Requester_id) REFERENCES Users(UserId)
);

CREATE TABLE Purchase_Order
(
  PO_Number INT NOT NULL,
  Date_Time DATE NOT NULL,
  Creator_id INT NOT NULL,
  Price_Quote INT NOT NULL,
  Required_by_Date DATE NOT NULL,
  Creator_UserId INT NOT NULL,
  PR_Number INT NOT NULL,
  Supplier_Id INT NOT NULL,
  Delivery_id INT NOT NULL,
  PRIMARY KEY (PO_Number),
  FOREIGN KEY (Creator_UserId) REFERENCES Users(UserId),
  FOREIGN KEY (PR_Number) REFERENCES Purchase_Requisition(PR_Number),
  FOREIGN KEY (Supplier_Id) REFERENCES Supplier(Id),
  FOREIGN KEY (Delivery_id) REFERENCES PO_Delivery_Details(Delivery_id)
);

CREATE TABLE Goods_Received
(
  PO_Id INT NOT NULL,
  GR_Id INT NOT NULL,
  DateTime DATE NOT NULL,
  PO_Number INT NOT NULL,
  GR_Delivery_Id INT NOT NULL,
  PRIMARY KEY (GR_Id),
  FOREIGN KEY (PO_Number) REFERENCES Purchase_Order(PO_Number),
  FOREIGN KEY (GR_Delivery_Id) REFERENCES
GR_Delivery_Recieved_Details(GR_Delivery_Id)
);
```

```
CREATE TABLE PO_Based_On_Contract
(
  Price_Rate NUMERIC NOT NULL,
  Contract_Id INT NOT NULL,
  PO_Number INT NOT NULL,
  PRIMARY KEY (Contract_Id, PO_Number),
  FOREIGN KEY (Contract_Id) REFERENCES Contracts(Id),
  FOREIGN KEY (PO_Number) REFERENCES Purchase_Order(PO_Number)
);

CREATE TABLE PR_Contains_Product
(
  Quantity INT NOT NULL,
  Remarks Text NOT NULL,
  PR_Number INT NOT NULL,
  Product_Id INT NOT NULL,
  PRIMARY KEY (PR_Number, Product_Id),
  FOREIGN KEY (PR_Number) REFERENCES Purchase_Requisition(PR_Number),
  FOREIGN KEY (Product_Id) REFERENCES Product(Id)
);

CREATE TABLE Product_Received_GR
(
  Quantity INT NOT NULL,
  Condition ENUM NOT NULL,
  Product_Id INT NOT NULL,
  GR_Id INT NOT NULL,
  PRIMARY KEY (Product_Id, GR_Id),
  FOREIGN KEY (Product_Id) REFERENCES Product(Id),
  FOREIGN KEY (GR_Id) REFERENCES Goods_Received(GR_Id)
);

CREATE TABLE PO_Contains_Product
(
  Quantity INT NOT NULL,
  Product_Id INT NOT NULL,
  PO_Number INT NOT NULL,
  PRIMARY KEY (Product_Id, PO_Number),
  FOREIGN KEY (Product_Id) REFERENCES Product(Id),
  FOREIGN KEY (PO_Number) REFERENCES Purchase_Order(PO_Number)
);

CREATE TABLE Purchase_Requisition_Justification
(
  Justification INT NOT NULL,
  PR_Number INT NOT NULL,
  PRIMARY KEY (PR_Number),
  FOREIGN KEY (PR_Number) REFERENCES Purchase_Requisition(PR_Number)
);
```

```
CREATE TABLE User_Contact_Details
(
  Email_Id VARCHAR(100) NOT NULL,
  Company_Address Text NOT NULL,
  City VARCHAR(100) NOT NULL,
  UserId INT NOT NULL,
  PRIMARY KEY (UserId),
  FOREIGN KEY (UserId) REFERENCES Users(UserId)
);

CREATE TABLE User_Phone_Numbers
(
  User_Phone_Number VARCHAR(15) NOT NULL,
  UserId INT NOT NULL,
  PRIMARY KEY (UserId),
  FOREIGN KEY (UserId) REFERENCES Users(UserId)
);

CREATE TABLE PR_Approver
(
  Comments INT NOT NULL,
  Approval_DateTime DATE NOT NULL,
  PR_Number INT NOT NULL,
  Approver_UserId INT NOT NULL,
  PRIMARY KEY (PR_Number, Approver_UserId),
  FOREIGN KEY (PR_Number) REFERENCES Purchase_Requisition(PR_Number),
  FOREIGN KEY (Approver_UserId) REFERENCES Users(UserId)
);

CREATE TABLE Contract_Product_Pricing
(
  Price_Rate INT NOT NULL,
  Discount_Rate INT NOT NULL,
  Product_Id INT NOT NULL,
  Contract_Id INT NOT NULL,
  PRIMARY KEY (Product_Id, Contract_Id),
  FOREIGN KEY (Product_Id) REFERENCES Product(Id),
  FOREIGN KEY (Contract_Id) REFERENCES Contracts(Id)
);

CREATE TABLE User_Dept_Approver
(
  Dept_Id INT NOT NULL,
  P_Budget_Approver INT NOT NULL,
  PRIMARY KEY (Dept_Id, P_Budget_Approver),
  FOREIGN KEY (Dept_Id) REFERENCES Department(Id),
  FOREIGN KEY (P_Budget_Approver) REFERENCES Users(UserId)
);

CREATE TABLE GR_Return_Details
(
  Quantity INT NOT NULL,
  Condition ENUM NOT NULL,
  Product_Id INT NOT NULL,
```

```
  GR_Id INT NOT NULL,
  PRIMARY KEY (Product_Id, GR_Id),
  FOREIGN KEY (Product_Id) REFERENCES Product(Id),
  FOREIGN KEY (GR_Id) REFERENCES Goods_Received(GR_Id)
);

CREATE TABLE Payment_Details
(
  Payment_Id INT NOT NULL,
  Budget_Code VARCHAR(10) NOT NULL,
  Refund_Amonut NUMERIC,
  GR_Id INT NOT NULL,
  PRIMARY KEY (Payment_Id, GR_Id),
  FOREIGN KEY (GR_Id) REFERENCES Goods_Received(GR_Id)
);

CREATE TABLE Payment_Details_Mode
(
  Mode INT NOT NULL,
  Amount NUMERIC NOT NULL,
  Details Text NOT NULL,
  Payment_Id INT NOT NULL,
  GR_Id INT NOT NULL,
  PRIMARY KEY (Payment_Id, GR_Id),
  FOREIGN KEY (Payment_Id, GR_Id) REFERENCES
Payment_Details(Payment_Id, GR_Id)
);
```

Sample Insert SQL Statements

```
INSERT INTO Department (Id, Name, Budget_Code) VALUES (1001, 'ENG',
'ENG0011');

INSERT INTO Roles (Id, RoleName) VALUES (1, 'Admin');

INSERT INTO Roles_Permissions(Role_id, Permissions) VALUES (1,
'READ'),(1, 'WRITE'), (1, 'MODIFY'), (1, 'DELETE');

INSERT INTO Users (UserName, UserId, Signature, Dept_Id, Role_Id) VALUES
('Braj Kishor', 1,
'12343dqewfr23qwefq32rwqdfq3wr23rfqwefqwefqw3rfqewfqwdf3', 1001, 1);


INSERT INTO User_Dept_Approver (Dept_Id, P_Budget_Approver) VALUES
(1001, 1);


INSERT INTO User_Phone_Numbers (UserId, User_Phone_Number) VALUES (1,
'+919223940394030'),(1, '+01133203');

INSERT INTO User_Contact_Details (UserID, Email_id, Company_Address,
City) VALUES (1, 'skdfsnf@officemail.com', '1/2, Mount Everest' 'MAR-
SARA');
```

```
INSERT INTO Purchase_Requisition (PR_Number, DateTime, Comments,
Requester_id) VALUES (123, now(), 'Raising PR', 2);

INSERT INTO Purchase_Requisition_Justification (PR_Number,
Justification) VALUES (123, 'Need Stuff');

INSERT INTO PR_Approver (PR_Number, Approver_UserId, Comments,
Approval_DateTime) VALUES (123, 1, 'Approved, now());

INSERT INTO Product (Id, Name, TYPE, Description) VALUES (1, 'MR
Chairs', 'OFFICE CONSUMABLE', 'Chairs FOR Office');

INSERT INTO PR_Containing_Product (PR_Number, Product_Id, Quantity,
Remark) VALUES (123, 1, 500, 'Needed Asap');


///////INSERTING THE VALUES IN THE ORDER OF THE COLUMNS//////


INSERT INTO Purchase_Order VALUES (1, Now(), 3, 25000.00, 3, 123, 1);

INSERT INTO PO_Delivery_Details VALUES (1, '91223213212', 'JAX', '31-
03-2020', 'dfsdfads@officemail.com');

INSERT INTO Suppliers VALUES (90001, 'Office Supplies', '01-01-2010',
'HIGH');

INSERT INTO Supplier_Contact_Details VALUES (90001, '23234234234',
'sdfasdf@suppliermail.com', 'MUMBAI');

INSERT INTO Contracts VALUES (3001, 'Valid Till 31-07-2020', 90001);

INSERT INTO Contract_Product_Pricing VALUES (1, 3001, '50.00', '5% per
100, 10% per 500');

INSERT INTO Goods_Received VALUES (1, 1, 1, 3220012);

INSERT INTO GR_Delivery_Recieved_Details VALUES (1, 'MAR-SARA',
'Tutood', Now());

INSERT INTO Payment_Details (3220012, 'ENG0011', 1, 332.00);

INSERT INTO Payment_Details_Mode ('DD', 3220012, 23000.00, 'DD No
12312312');
```

Sample Update SQL Statements

Update Primary Budge Approver for Departments

```
UPDATE User_Dept_Approver SET P_Budget_Approver = 5 WHERE Dept_Id = 1002;
```

Update Contract Terms and Pricing after negotiation with Supplier

```
UPDATE Contracts SET Contract_Terms = 'New Contract Term' WHERE Id = 90001;
UPDATE Contract_Product_Pricing SET Price_Rate = '70', Discount_Rate = '10% per 100, 20% per 1000' WHERE Contract_Id = 90001 AND Product_Id = 1;
```

Update PO Delivery Details Incase supplier needs to reschedule delivery

```
UPDATE PO_Delivery_Details SET Location = 'New Location', Receiver_Email = 'new email', Required_By_DateTime = 'new date', Receipient_Contact_Number = '123123123' WHERE PO_Number = 1;
```

Update Supplier Rating incase supplier last few deliveries were not good.

```
UPDATE Supplier SET Rating = 'Average' WHERE Id = 3001;
```

Update Refund Amount in case products were returned upon Goods Received

```
UPDATE Payment_Details SET Refund_Amount = 400.00 WHERE Payment_Id = 123123 AND GR_Id = 2;
```

Sample Select SQL Queries

Select all the PR details which have been approved which has been raised by users of a given department.

```
SELECT * FROM Purchase_Requisition as PR INNER JOIN Purchase_Requisiton_Justification as PRJ ON PR.PR_Number = PRJ.PR_Number INNER JOIN PR_Approver as PRA ON PR.Number = PRA.Number INNER JOIN Users as U ON U.User_Id = PRA.Approver_UserId INNER JOIN Department as D ON U.Dept_Id = D.Id WHERE D.Id = 1004;
```

For a given User, Show all his PRs and POs

```
SELECT * FROM User U INNER JOIN Purchase_Requisition PR ON PR.Requester_UserId = U.UserId INNER JOIN Purchase_Order PO ON PR.PR_Number = PO.PR_Number WHERE U.UserId = 'Csssld';
```

Select all POs associated with a given PR.

```
SELECT * FROM Purchase_Requisition as PR INNER JOIN Purchase_Order as PO ON PR.PR_Number = PO.PR_Number INNER JOIN PO_Delivery_Details as POD ON PO.PO_Number = POD.PO_Number WHERE PR.PR_Number = 1231;
```

For A given PR, what was the total quantity which was delivered vs the quantity which was ordered for each product?

```
SELECT PR.PR_Number, PRCP.Product_Id as 'Product Ordered', PRCP.Quantity
'Quantity Ordered', PGR.Product_Id as 'Product Received', PGR.Quantity -
GRD.Quantity as 'Quantity Received'  FROM Purchase_Requisition as PR INNER
JOIN PR_Contains_Product as PRCP ON PRCP.PR_Number = PR.PR_NUMBER INNER JOIN
Product as P on P.Id = PRCP.Purchase_Order as PO ON PR.PR_Number =
PO.PR_Number INNER JOIN Goods_Received as GR ON PO.PO_Number = GR.PONumber
INNER JOIN Product_Received_GR as PGR ON PGR.GR_Id = GR.GR_Id AND P.Id =
PGR.Product_Id INNER JOIN GR_Return_Details as GRD on GRD.GR_Id = GR.GR_Id
AND P.Id = GRD.Product_Id WHERE PR.PR_Number = 123;
```

Select the Best supplier for each category who has given maximum discount till now.

```
SELECT Id, Category, MAX(Discount_Sum), Category FROM (SELECT SUM(Discount)
as Discount_Sum, Supplier.Id FROM Supplier INNER JOIN Receipt ON
Supplier.Id = Receipt.Id GROUP BY Supplier.Id) WHERE Category = 'HIGH'
GROUP BY Id, Category;
```

For each product, how many contracts and suppliers are present?

```
SELECT P.Id, P.Name, COUNT(Contract_Id), Count(Supplier_Id) FROM Product as
P LEFT OUTER JOIN Contract_Product_Pricing as CPP ON P.Id = CPP.Product_Id
INNER JOIN Contracts as C ON CPP.Contract_Id = C.Id GROUP BY p.Id, P.Name;
```

For each department what is the Average Order Price in the sorted order descending?

```
SELECT D.Id, D.Name, Avg(Price_Quote) FROM Department D LEFT OUTER JOIN
User U ON D.Id = U.DeptId INNER JOIN Purchase_Requisition PR ON U.UserId =
PR.Requestor_id INNER JOIN Purchase_Order PO ON PR.PR_Number = PO.PR_Number
GROUP BY D.Id, D.Name ORDER BY Avg(Price_Quote) DESC;
```

For a given supplier which is the most preferred mode of payment?

```
SELECT Id, Max(Mode_Count), Mode as 'Preferred Mode' FROM (SELECT S.Id as
Supplier, PDM.COUNT(Mode) as Mode_Count, Mode FROM Supplier S INNER JOIN
Purchase_Order PO ON S.Id = PO.Supplier_Id INNER JOIN Goods_Received GR ON
PO.PO_Number = GR.PO_Number INNER JOIN Payment_Details PD ON PD.GR_Id =
GR.GR_Id INNER JOIN Payment_Details_Mode PDM ON PDM.GR_Id = PD.GR_Id AND
PD.Payment_Id = PDM.Payment_Id GROUP BY S.Id, Mode ORDER BY
PDM.Count(Mode) DESC) GROUP BY Id, Mode;
```

For each supplier for whom a Purchase Order was opened, how many times, he delivered product which were not required to be returned and what was the quality of the product?

```
SELECT S.Id, Count(GR.Id), Condition FROM Supplier S INNER JOIN
Purchase_Order PO ON S.Id = PO.Supplier_Id INNER JOIN Goods_Received GR ON
PO.PO_Number = GR.PO_Number INNER JOIN Product_Received_GR PRGR ON
PRGR.GR_Id = GR.GR_Id WHERE GR.Id NOT IN (SELECT GR_ID FROM
GR_Return_Details) GROUP BY S.Id, Condition;
```

Indices On the Tables

Following Indices are present on the tables.

Key Attribute Indices

Primary Keys

Each table is having it's own primary key. Primary key selected is naturally a unique Identifier column which has been selected for each of the tables.

For the weak entity tables their primary key's as the Combination of the weak entities unique key and the foreign key of the parent entity.

For the table which are made for Associative entities has, the primary key is the combination of both the foreign keys to the related entities.

Unique Keys

Additionally, following entities have Unique keys present

1. Department – **BudgeCode** – BudgetCode is unique for each dept and can be used to for finance reporting and auditing to figure out Department's expenses.
2. Roles – **RoleName** – Each user has a single rolename and rolename is always unique.
3. PO_Delivery_Details
    a. **Receipient_Email** – Can be used for fast searching a contact for a given location.
4. GR_Delivery_Received_Details:
    a. **Location** – Can be used for fast searching where the Product was delivered.

Non-Key Indices

ENUMS

ENUM Datatype when used, creates its own type of **Index**. ENUMS can be used when a column has only specified set of values possible. The ENUMS are String representation but has Integer Storage and Indexing Values.

Following Colums are Declared ENUMS

1. Role -**RoleName** – The role names are fixed. There can only be certain types. Declaring this ENUM can speed up queries related to User Roles, Access and Permissions.

2. Product_Received_GR - **Condition** - The condition of products received. Only a few key conditions are possible. With ENUM, searches related to Product quality received, supplier ratings etc can become faster.
3. GR_Return_Details – **Condition** – The condition of products returned. With ENUM, queries related what was returned and why, Supplier Rating, Refund Calculation etc will become faster.
4. Product – **Type** – There are fixed types of Product possible. Declaring this ENUM can be used to group the Product when Showing in UI and Drill Down on those types.
5. Supplier – **Category** – The category of product which supplier delivers. This being ENUM can be used to Group Suppliers and Showing then UI, Faster Drill Down, Faster searches while comparing multiple suppliers who supply same type of product

Additional Non-Key Indices which can Declared.

Apart from all the indices which already have been declared above, following additional Indices can become helpful in certain scenarios.

1. Product – **Name** – A **Full Text Search Index** can be created on this field to speed up searching of the products especially if the List of Products is too large.
2. Purchase_Requestion - **DateTime** – An Index on this column can speed up queries related to sorting the PR's raised and Approved.
3. Purchase_Order – **DateTime** – An Index on this can speed up queries related to sorting the PO's raised and Deliveries and Turn Around Times.
4. Supplier-Contact-Details – **Location** – An Index on this can speed up queries to find a supplier in a given location.