

**DATABASE**

**SPECIFICATIONS**

*eCommerce store for Penn State Great Valley*

*Team 2*

**School of Graduate Professional Studies**

Information Science Department

IN SC 521 - Introduction to Database Concepts

August, 2018

# Document Control

## Work carried out by:

|  |  |  |
| --- | --- | --- |
| **Name** | **Email Address** | **Other** |
| Deni Avinash Aalla Balaji | Dua444@psu.edu |  |
| Utsav Goel | Ujg5001@psu.edu |  |
|  |  |  |

## Revision Sheet

|  |  |  |
| --- | --- | --- |
| **Release No.** | **Date** | **Revision Description** |
| 1 | 08/27/2018 | Purpose and Overview |
| 2 | 08/31/2018 | ER Diagram |
| 3 | 09/13/2018 | Data Dictionary |
| 4 | 09/20/2018 | Relational Schema |
| 5 | 09/26/2018 | Table Schemas |
| 6 | 10/03/2018 | Tables Normal Form |
| 7 | 10/07/2018 | Select Queries |

**DATABASE SPECIFICATIONS**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| S.No | Contents | Page.No |
| 1 | Purpose | 1 |
| 2 | System Overview | 1 |
| 3 | ER Diagram | 2 |
| 4 | Data Dictionary | 4 |
| 5 | Relational Schema | 7 |
| 6 | Table Schema | 8 |
| 7 | Table Normal Form | 18 |
| 8 | Select Queries0 | 29 |

## Purpose

## The purpose of this project is to design a database for the proposed eCommerce store for Penn State Great Valley. The System will be used to support an eCommerce website to allow users to buy or reserve articles for future purchases from the Convenience store on the first floor of the Penn State Great Valley campus.

## System Overview

1. System name or title: Penn State Great Valley Convenience Store Database Management System
2. Core Requirements :

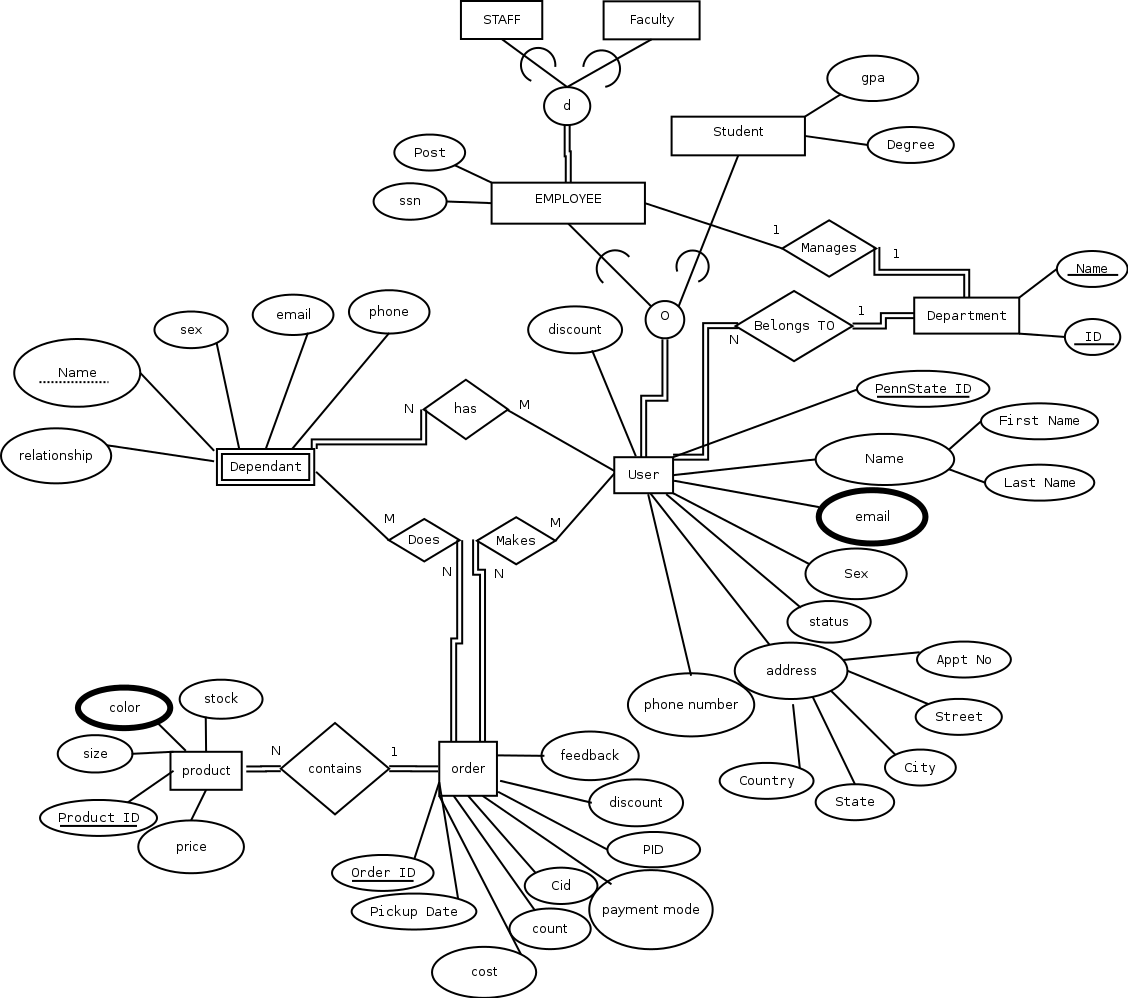
* The system will allow users to make purchases from the store with options for delivery of said items or pickup option so that users may pick up items directly from the store themselves.
* The proposed users will include Students, Staff, Faculty and dependents of all the above. Users will also be allowed to provide feedback about the order.
* Each user will have an ID associated with them i.e. the Penn State ID. Staff and students will be considered different entities so that they can be offered different discounts depending on their position. Dependents will be associated with the respective Entities which allow them to access the Penn State services.
* Some attributes will be common for both Staff and students i.e. First name, last name, Penn State ID, Account Status, Telephone number, email address and Sex.
* Staff will also have Post, Discount % and Department. Each post in a department will have the same discount percentage associated with it.
* Students will have GPA, Discount %, Degree and Department associated with their accounts. Discounts will be allotted to Students on basis of GPA, Students with same GPA will have same discounts.
* Dependents will be identified based on their own name and the corresponding user’s Penn State ID.
* In case a dependent can be associated with more than a single member of the Penn State Community, we only consider the highest association i.e. if a dependent is related to both a staff and a student, we only associate him/her with the Staff as he/she gets better benefits with it.
* Every User belongs to a department and each department is managed by an employee.
* The store will also maintain a Product table which lists the inventory of the store. Each product will have a productID associated with it. Each product will also have the following details: Size, Color, Price and Number of Items in inventory.
* There will also be an Orders table which will store all the order details of any purchases or orders made on the store. Each order will have an OrderID, CustomerID and will contain the number, the ProductIDs of the products purchased, the total billing amount, Discount given, mode of payment (Cash, debit card or credit card) and expected pickup date. Only a single CustomerID may be used on a purchase.
* Table Standards :
* All the tables will be normalized up to the third normal form.
* Primary keys such as the PennStateIDs, DependantID, OrderID, ProductID will be used to uniquely identify the records.
* Foreign keys should be employed to maintain referential integrity.
* Any attributes which are required to be added for reference will also be added later as the workflow is refined.

## ER Diagram

**Assumptions and Constraints**:

* Unique Keys cannot be represented in ERD.
* Following Attributes are implied : **ID, Name, email, sex, discount, address, ph number, status, department, degree, Gpa,SSN, post, order id, customer id, count, cost, pickup date, payment mode, product id, feedback, size, colour, price , stock, Department name and ID.**
* Following Entities are implied : **Users ( Employee{Staff, Faculty}, Students), Dependent, Order, Product, Department**
* Following relationships are implied (Relationships are in block letters):
* User may or not may not **have** one or more dependents.
* User may or may not **make** Orders
* A User can make multiple Orders
* Dependent may or may not **do** Orders
* A Dependent can make multiple orders
* Users are further divided into subsystems like Employee and student.
* Employee is divided into subsystems like staff and faculty.
* Users **belong** to a department
* Each department is **managed** by an Employee
* Every Order **contains** at least a Product

**ER Diagram 1**



**Data Dictionary**

**Dependent Table:**

**Description** : A dependent is one or person who do not work at Penn State but are related to a Penn State Staff of student.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable | Variable Name | Variable Type | Values | Notes |
| Dependent | Name | String | Deni Utsav | Partial Key |
| Relationship | Relationship Type | Numeric | 1 – Spouse  2 – Child  3 – Domestic Partner |  |
| Sex | Sex | String | Male or Female |  |
| Email | Email | String | deniutsav@gmail.com |  |
| Phone | Phone | (###) ###-#### | (610) 504-7777 |  |

**Product Table :**

**Description** : Product contains all details of an individual item or a product.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable | Variable Name | Variable Type | Values | Notes |
| Product ID | PID | Numeric | 1111 | Primary Key |
| Size | Size | Numeric | 10 |  |
| Colour | Colour | Numeric | 0 – NA  1 – Violet  2 – Indigo  3 – Blue  4 – Green  5 – Red  6 – Yellow  7 - Orange | Can be muli-valued with a single product having to multiple colours |
| Price | Price | Float Numeric | 4.5 | Values can be decimal and all are in dollars. |
| Stock | Stock | Numeric | 5 |  |

**Order Table :**

**Description :** Contains all the order details for with the users details and product details for every order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable | Variable Name | Variable Type | Values | Notes |
| Order ID | OID | Numeric | 22222 | Primary Key |
| Pickup Date | Pickup Date | YYYYMMDD | 20180912 | Standard Date format |
| Cost | Cost | Float Numeric | 5.5 | Values can be decimal and all are in dollars. |
| Count | Count | Numeric | 3 |  |
| CID | CID | Alpha Numeirc | 01Deni99887766  Or 0299887766 | There are two possibilities where Customer ID can be dependent or actual users. Dependents are identified using the first two digits ‘01’ and users by ‘02’ then the name and User id follows for customer and only user id for actual users. |
| Payment Mode | Payment Mode | Numeric | 1. Debit Card 2. Credit Card 3. Net banking |  |
| PID | PID | Numeric | 1111 | Same as Product ID in Product table |
| Discount | Discount | Numeric | 10 | Discount is denoted using numbers in percentage |
| Feedback | Feedback | String | Good Product | Can be of any length String |

**Employee Table**

**Description:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Variable Name** | **Variable Type** | **Values** | **Notes** |
| Post | Post | String | Various Posts in the Department, may vary according to Staff or Faculty (Associate Professor, Assistant Professor etc.) | Describes post held by individual |
| **Ssn** | **Ssn** | String | ######### | 9 letter numeric string constrained by regular expressions. |
| Type | Type | String | Faculty/Staff | Instead of defining a separate table for Staff and faculty, we decided to put both in the same table Employee and give a separate attribute for type. |

**Student Table**

**Description:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Variable Name** | **Variable Type** | **Values** | **Notes** |
| GPA | Gpa | Float numeric | 0.0-4.0 | To record student’s upheld CGPA |
| Degree | degree | String | B.S. , M.S. | To differentiate between students studying Masters Degrees and Bachelor’s Degrees. |

**User Table**

**Description:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Variable Name** | **Variable Type** | **Values** | **Notes** |
| Penn State ID | PSU-ID | String | Any Alpha numeric string constrained by Regular Expressions | Primary Key |
| First Name | F-Name | String | Utsav |  |
| Last Name | L-Name | String | Goel |  |
| Sex | Sex | String | Male/Female | Can only be either Male or Female |
| Status | Status | String | Full-Time/Part-Time | To differentiate between full-time students and part-time students |
| Appt No | Addr\_Apt\_no | String | Unit 223 | Instead of storing the composite attribute in another table, we can store the various values in |
| Street | Addr\_Strt | String | Creekside Lane |
| City | Addr\_City | String | Malvern |
| State | Addr\_State | String | Pennsylvania |
| Country | Addr\_Country | String | United States of America |
| Phone Number | Phn\_num | String | (###)-###-#### |  |
| Discount | discount | Float Numeric | 4.5/6.0/8.0 | Discount depending on whether the user is a student, staff or employee. |

**Department Table**

**Description:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Variable Name** | **Variable Type** | **Values** | **Notes** |
| Name | Dept\_Name | String | Engineering/Management | Primary Key |
| ID | Dept\_ID | String | Aplhanumeric string to differentiate departments | Primary Key |

**HOD Table**

**Description:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Variable Name** | **Variable Type** | **Values** | **Notes** |
| PSU ID | PSU ID of HOD | String | 999999999 | Primary key |
| ID | Dept\_ID | String | Aplhanumeric string to differentiate departments |  |

**Relational Schema**

User(First Name, Last Name,Sex,Status, Appt No, Street, City, State, Country, Phone Number,discount)   
PK : PennState ID  
FK : None  
  
User\_EMAIL()  
PK : {PennState ID, email}  
FK : PennState ID of User table  
  
STUDENT(Gpa,degree)   
PK : PennState ID  
FK : {Department Name , ID}, PennState ID

EMPLOYEE(ssn,post,type, Department ID)

PK : PennStateID

FK : PennStateID of user  
  
DEPARTMENT(Name)  
PK : {ID}  
FK : ID

HOD(Department ID)

PK: PSU\_ID

FK: Department ID  
  
DEPENDENT(relationship, sex,email,phone)  
PK : {Name,PennState ID}  
FK : User PSU\_ID  
  
  
ORDER(Cid,Count,Payment Mode,PID, cost, discount, feedback, Pickup date)  
PK : Order ID  
FK : Order ID of contains  
  
USER\_ORDERS()  
PK : {PennStateID, order ID}  
FK : PennStateID, Order ID  
  
DEPENDENT ORDERS()  
PK : {order ID, Dependent Name, PennState ID}  
FK : order ID, {Dependent Name, PennState ID}  
  
PRODUCT(price,size,stock)  
PK : Product ID  
FK : Product ID of contains

O\_CONTAINS()

PK: {Order ID,Product ID}

FK: Order ID,Product ID  
  
COLOR()  
PK : {Product ID,color}  
FK : Product ID of product

## ****Tables schemas:****

|  |  |  |  |
| --- | --- | --- | --- |
| **User\_PSU** |  |  |  |
| **Description** | This tables contains all common user details. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **PSU\_ID** | Penn State ID of a student | String | 992233447 |
| **F\_Name** | First Name of a student | String | Deni Avinash |
| **L\_Name** | Last Name of a student | String | Aalla Balaji |
| **Sex** | Gender of a student | String | M |
| **Status** | Status of a student | String | Active |
| **Apt\_num** | Apartment number of a student | String | Unit 223 |
| **Street** | Street name from student address | String | Creekside Lane |
| **City** | City name from student address | String | Malvern |
| **State** | State name from student address | String | Pennsylvania |
| **Country** | Country of residence | String | United Sates |
| **Phn\_num** | Phone number of a student | Integer | 1234567890 |
| **Discount** | Discount percentage rate for a student | float | 8.0 |
| **Primary Key** | PSU\_ID | | |
| **Foreign Keys** |  | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.User\_PSU(  PSU\_ID char(9) PRIMARY KEY,  F\_Name varchar2(10) NOT NULL,  L\_Name varchar2(10) NOT NULL,  Sex char NOT NULL,  status varchar2(8) NOT NULL,  Apt\_num varchar2(5) NOT NULL,  street varchar2(10) NOT NULL,  city varchar2(9) NOT NULL,  state char(2) NOT NULL,  country varchar2(10) NOT NULL,  phn\_num char(10) NOT NULL,  discount real NOT NULL,  check (discount >= 0.0 and discount < 10.0),  check (Sex in ('m','f')),  check (status in ('active','inactive'))  );  INSERT INTO User\_PSU values('999999999','Utsav','Goel','m','active','223','45Creeksid','Malvern','PA','USA','4843189042',6.0);  INSERT INTO User\_PSU values('999999998','Deni','Avinash','m','active','223','45Creeksid','Malvern','PA','USA','4843189041',6.0);  INSERT INTO User\_PSU values('999999997','Aaqil','Rahman','m','active','223','45Creeksid','Malvern','PA','USA','4843189040',6.0);  INSERT INTO User\_PSU values('999999967','Kavin','Rahman','m','active','112','Exton St.','Malvern','PA','USA','4843189030',6.0);  INSERT INTO User\_PSU values('999999957','Acchyuta','Goel','m','active','211','Paoli St.','Malvern','PA','USA','4843189020',6.0);  INSERT INTO User\_PSU values('999999947','Dhanjeet','Vardhan','m','active','299','West St.','Malvern','PA','USA','4843189010',6.0); | | |
| **Count of records in the table** | 6 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **User\_PSU\_Email** |  |  |  |
| **Description** | This table contains the Email address of the students and employees. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **Email** | **Email of the user** | **String** | Dua444@psu.edu |
| **PSU\_ID** | **Penn State ID od user** | **String** | 992233447 |
| **Primary Key** | {PSU\_ID, Email} | | |
| **Foreign Keys** | PSU\_ID | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.User\_PSU\_email(  PSU\_ID char(9) NOT NULL,  email varchar2(20) NOT NULL,  PRIMARY KEY(PSU\_ID,email),  FOREIGN KEY (PSU\_ID) REFERENCES User\_PSU(PSU\_ID)  );  INSERT into User\_PSU\_email values('999999999','utsav96@gmail.com');  INSERT INTO user\_psu\_email values('999999999','ujg5001@psu.edu');  INSERT into User\_PSU\_email values('999999998','deni@gmail.com'); | | |
| **Count of records in the table** | 3 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Employee** |  |  |  |
| **Description** | This table contains all the employee details. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **PSU\_ID** | **Penn State ID of an Employee** | **String** | 334466775 |
| **SSN** | **SSN of an Employee** | **Integer** | 112233445 |
| **Post** | **Position of an Employee** | **String** | Assistant Professor |
| **Type** | Check if it is a staff | Char | Y |
| **Dept\_ID** | ID of employee department | String | 3 |
| **Primary Key** | PSU\_ID | | |
| **Foreign Keys** | PSU\_ID | | |
| **SQL Code** | create table8i USERINSC521FA18\_UJG5001.Employee(  ssn char(9) UNIQUE NOT NULL,  post varchar2(10) NOT NULL,  PSU\_ID char(9) PRIMARY KEY,  Dept\_ID char(5) NOT NULL,  type varchar2(7) NOT NULL,  FOREIGN KEY (PSU\_ID) references User\_PSU(PSU\_ID),  check (type in ('faculty','staff'))  );  INSERT into employee values('123456789','prof','999999999','5001','faculty');  INSERT Into employee values('123456788','prof','999999998','5003','faculty');  INSERT Into employee values('123456787','clerk','999999997','5001','staff'); | | |
| **Count of records in the table** | 3 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** |  |  |  |
| **Description** | This table contains all the Student details. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **PSU\_ID** | **Penn State ID of an Employee** | **String** | 334466775 |
| **GPA** | **GPA of a student** | **float** | 3.5 |
| **Degree** | **Degree of a student** | **String** | MS in Data Analytics |
| **Dept\_ID** | Department ID student belongs to | String | 3 |
| **Primary Key** | PSU\_ID | | |
| **Foreign Keys** | Dept\_ID, PSU\_ID | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.Student(  PSU\_ID char(9) PRIMARY KEY,  GPA real,  Degree varchar2(10) NOT NULL,  Dept\_ID char(5) NOT NULL,  FOREIGN KEY (PSU\_ID) references User\_PSU(PSU\_ID),  FOREIGN KEY (Dept\_ID) references Department(Dept\_ID),  check (Degree in ('bachelor','master','phd'))  );  insert into student values('999999967',4.0,'bachelor','50101');  insert into student values('999999957',4.0,'master','50101');  insert into student values('999999947',4.0,'phd','50103'); | | |
| **Count of records in the table** | 3 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Department** |  |  |  |
| **Description** | This table contains the details about a department. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **Dept\_ID** | **ID of department** | **Integer** | 5 |
| **Dept\_Name** | **Name od department** | **String** | Engineering |
| **Primary Key** | ID | | |
| **Foreign Keys** |  | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.Department(  Dept\_name varchar2(10) NOT NULL UNIQUE,  Dept\_ID char(5) PRIMARY KEY,  );  insert into department values('DaAn', '50101');  insert into department values('INSC','50103'); | | |
| **Count of records in the table** | 2 |  |  |

|  |  |  |
| --- | --- | --- |
| **HOD** |  |  |
| **Description** | This table contains the details about HOD of a department. | |
| **Attribute** | **Description** | **Type** |
| **Dept\_ID** | **ID of department** | **String** |
| **PSU\_ID** | **ID of HOD** | **String** |
| **Primary Key** | ID | |
| **Foreign Keys** | Dept\_ID | |
| **SQL Code** | Create table USERINSC521FA18\_DUA444.HOD  (  PSU\_ID char(9) primary key,  Dept\_ID char(5) unique,  FOREIGN KEY (Dept\_ID) REFERENCES USERINSC521FA18\_DUA444.Department(Dept\_ID)  );  insert into USERINSC521FA18\_DUA444.HOD values ('999999999','50101');  insert into USERINSC521FA18\_DUA444.HOD values ('999999998','50103'); | |
| **Count of records in the table** | 2 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Dependant** |  |  |  |
| **Description** | This table contains all the details about dependents. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **Name** | **Name of dependent** | **String** | Utsav Goel |
| **Relationship** | **Relationship type for a student or an employee** | **String** | Child |
| **Sex** | **Gender of the dependent** | **String** | M |
| **Email** | **Email address of dependent** | **String** | abc@gmail.com |
| **Phn\_Num** | **Phone number of dependent** | **Integer** | 2233445561 |
| **PSU\_ID** | **ID of user** | **String** | 99445576 |
| **Primary Key** | {PSU\_ID, Name} | | |
| **Foreign Keys** | PSU\_ID of user | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.Dependant(  PSU\_ID char(9) NOT NULL,  Name varchar2(20) NOT NULL,  email varchar2(30) NOT NULL,  Sex char NOT NULL,  phn\_num char(10) NOT NULL,  relationship varchar2(10) NOT NULL,  PRIMARY KEY (PSU\_ID,Name),  FOREIGN KEY (PSU\_ID) references User\_PSU(PSU\_ID),  check (Sex in ('m','f'))  );  INSERT INTO Dependant values ('999999999','RAJ','abcdefg@gmail.com','m','4841221324','son');  INSERT INTO Dependant values ('999999998','Sharmila','sharmila.balaji@gmail.com','f','4841221325','mother');  INSERT INTO Dependant values ('999999997','Aaqil Jr.','redmpsdm@gmail.com','m','4841221328','son'); | | |
| **Count of records in the table** | 3 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Order\_t** |  |  |  |
| **Description** | This table contains all the order details. | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **O\_ID** | **ID of the order** | **String** | Abc12345 |
| **C\_ID** | **Customer ID** | **Integer** | 99445678 |
| **Pay\_Mode** | **Payment method to do payment** | **String** | Debit Card |
| **Discount** | **Discount price for the user** | **float** | 7.5 |
| **Feed\_back** | **Feedback given by customer** | **String** | Good Product |
| **Pickup\_Date** | **Expected Date of pickup from store** | **Date** | 10/10/2018 |
| **Cost\_O** | **Cost of the order** | **float** | 10.5 |
| **Count\_P** | **Count of products in order** | **Integer** | 3 |
| **Primary Key** | O\_ID | | |
| **Foreign Keys** | O\_ID of contains | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.Order\_t(  Feed\_back varchar2(20),  Discount real NOT NULL,  Pickup\_date Date,  cost\_o real,  count\_p number,  pay\_mode char(4),  C\_ID char(9),  O\_ID char(10) PRIMARY KEY,  FOREIGN KEY (O\_ID) references O\_Contains(O\_ID),  check (count\_p>0),  check (cost\_o>0)  );  insert into order\_t values('',6.0,'28-10-18',18.79,1,'cash','999999967','C\_11111111');  insert into order\_t values('',6.0,'28-10-18',18.79,1,'cash','999999947','C\_11111112');  insert into order\_t values('',6.0,'28-10-18',18.79,1,'cash','999999947','C\_11111113');  insert into order\_t values('',6.0,'28-10-18',18.79,1,'cash','999999957','C\_11111114'); | | |
| **Count of records in the table** | 4 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **User\_PSU\_Orders** |  |  |  |
| **Description** | This table links user and order | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **O\_ID** | **ID for the order** | **String** | Abc1234 |
| **PSU\_ID** | **Penn State ID of user** | **Integer** | 992233447 |
| **Primary Key** | {PSU\_ID, O\_ID} | | |
| **Foreign Keys** | PSU\_ID, O\_ID | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.User\_PSU\_orders(  O\_ID char(10) NOT NULL,  PSU\_ID char(9) NOT NULL,  PRIMARY KEY (O\_ID,PSU\_ID),  FOREIGN KEY (O\_ID) REFERENCES Order\_t(O\_ID),  FOREIGN KEY (PSU\_ID) REFERENCES User\_PSU(PSU\_ID)  );  INSERT INTO User\_PSU\_orders values ('C\_11111113','999999999');  INSERT INTO User\_PSU\_orders values ('C\_11111114','999999998'); | | |
| **Count of records in the table** | 2 |  |  |

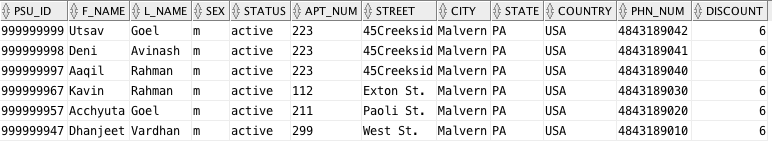
|  |  |  |  |
| --- | --- | --- | --- |
| **Dependent\_Orders** |  |  |  |
| **Description** | This table links user and order | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **O\_ID** | **ID for the order** | **String** | Abc1234 |
| **PSU\_ID** | **Penn State ID of user** | **Integer** | 992233447 |
| **Name** | **Name of Dependent** | **String** | Utsav Goel |
| **Primary Key** | { O\_ID, Name,PSU\_ID } | | |
| **Foreign Keys** | O\_ID, { Name,PSU\_ID} | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.Dependent\_orders(  O\_ID char(10) NOT NULL,  Name varchar2(20) NOT NULL,  PSU\_ID char(9) NOT NULL,  PRIMARY KEY (O\_ID, Name, PSU\_ID),  FOREIGN KEY (O\_ID) references Order\_t(O\_ID),  FOREIGN KEY (PSU\_ID,Name) references Dependant(PSU\_ID,Name)  );  INSERT INTO Dependent\_orders values ('C\_11111111','RAJ','999999999');  INSERT INTO Dependent\_orders values ('C\_11111112','Sharmila','999999998'); | | |
| **Count of records in the table** | 2 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Product** |  |  |  |
| **Description** | This table contains all the details about the products | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **P\_ID** | **ID for the product** | **String** | 1234567 |
| **Price** | **Price of the product** | **Float** | 13.5 |
| **Size\_p** | **Size of the product** | **Float** | 10.5 |
| **Stock** | **Available stock count** | **Integer** | 50 |
| **Primary Key** | P\_ID | | |
| **Foreign Keys** | P\_ID of contains | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.Product(  price real NOT NULL,  size\_p char(2) NOT NULL,  stock number NOT NULL,  p\_ID char(7) PRIMARY KEY,  check (stock>0),  check (size\_p in ('xs','s','m','l','xl')),  check (price >=0.0)  );  Alter table Product ADD FOREIGN KEY (P\_ID) references O\_contains(P\_ID);  insert into product values(19.99,'xs',20,'H\_543');  insert into product values(19.99,'s',20,'H\_544');  insert into product values(29.99,'xl',20,'H\_546');  insert into product values(24.99,'l',20,'H\_545'); | | |
| **Count of records in the table** | 4 |  |  |

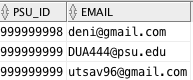
|  |  |  |  |
| --- | --- | --- | --- |
| **O\_Contains** |  |  |  |
| **Description** | This table contains color code for products | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **P\_ID** | **ID for the product** | **String** | **Abc123** |
| **O\_ID** | **ID of order** | **String** | **123abc** |
| **Primary Key** | {P\_ID, O\_ID} | | |
| **Foreign Keys** | P\_ID , O\_ID | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.O\_Contains(  O\_ID char(10) NOT NULL UNIQUE,  p\_ID char(7) NOT NULL UNIQUE,  PRIMARY KEY (O\_ID, p\_ID),  FOREIGN KEY (p\_ID) references Product(p\_ID)  );  Alter table O\_contains ADD FOREIGN KEY (O\_ID) references Order\_t(O\_ID);  insert into o\_contains values('C\_11111111','H\_544');  insert into o\_contains values('C\_11111112','H\_543');  insert into o\_contains values('C\_11111113','H\_546');  insert into o\_contains values('C\_11111114','H\_545'); | | |
| **Count of records in the table** | 4 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Color** |  |  |  |
| **Description** | This table contains color code for products | | |
| **Attribute** | **Description** | **Type** | Examples of values |
| **P\_ID** | **ID for the product** | **String** | **Abc123** |
| **Color** | **Color of product** | **String** | **Blue** |
| **Primary Key** | P\_ID,Color | | |
| **Foreign Keys** | P\_ID of product | | |
| **SQL Code** | create table USERINSC521FA18\_UJG5001.Color(  color varchar2(10) NOT NULL,  p\_ID char (7) NOT NULL,  PRIMARY KEY (p\_ID,color),  FOREIGN KEY (p\_ID) REFERENCES Product(p\_ID)  );  insert into Color values('Blue', 'H\_544');  insert into Color values('Red', 'H\_543');  insert into Color values('Green', 'H\_545');  insert into Color values('Yellow', 'H\_546'); | | |
| **Count of records in the table** | 4 |  |  |

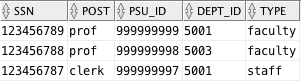
## ****Tables Normal Form****



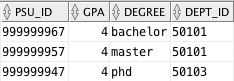
|  |  |  |
| --- | --- | --- |
| **User\_PSU** |  |  |
| **Functional Dependencies** | PSU\_ID -> {F\_Name, L\_name, Sex, status,apt\_num, street,city,state,country,phn\_num,discount} | |
| **Primary Key** | PSU\_ID | PSU\_ID uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



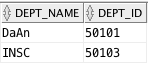
|  |  |  |
| --- | --- | --- |
| **User\_PSU\_Email** |  |  |
| **Functional Dependencies** | {PSU\_ID,Email} -> {Email} | |
| **Primary Key** | {PSU\_ID, Email} | {PSU\_ID, Email} uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



|  |  |  |
| --- | --- | --- |
| **Employee** |  |  |
| **Functional Dependencies** | PSU\_ID -> {SSN,Post,Dept\_ID,Type} | |
| **Primary Key** | PSU\_ID | PSU\_ID uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



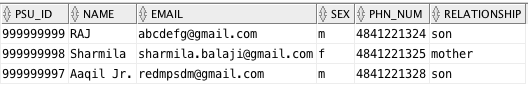
|  |  |  |
| --- | --- | --- |
| **Student** |  |  |
| **Functional Dependencies** | PSU\_ID -> {GPA,Degree,Dept\_ID} | |
| **Primary Key** | PSU\_ID | PSU\_ID uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



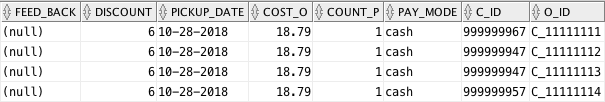
|  |  |  |
| --- | --- | --- |
| **Department** |  |  |
| **Functional Dependencies** | Dept\_ID -> Dept\_Name | |
| **Primary Key** | Dept\_ID | Dept\_ID uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



|  |  |  |
| --- | --- | --- |
| **HOD** |  |  |
| **Functional Dependencies** | PSU\_ID -> Dept\_ID | |
| **Primary Key** | PSU\_ID | PSU\_ID uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



|  |  |  |
| --- | --- | --- |
| **Dependent** |  |  |
| **Functional Dependencies** | {PSU\_ID,Name} -> {Email,Sex, Phn\_num, relationship} | |
| **Primary Key** | {PSU\_ID, Name} | {PSU\_ID, Name} uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



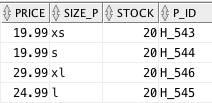
|  |  |  |
| --- | --- | --- |
| **Order\_t** |  |  |
| **Functional Dependencies** | O\_ID ->{Feedback,discount,pickup\_date,cost\_o,count\_p,pay\_mode,c\_id} | |
| **Primary Key** | O\_ID | O\_ID uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



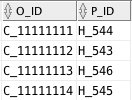
|  |  |  |
| --- | --- | --- |
| **User\_PSU\_Orders** |  |  |
| **Functional Dependencies** | {PSU\_ID, O\_ID}->{PSU\_ID, O\_ID} | |
| **Primary Key** | {PSU\_ID, O\_ID} | {PSU\_ID, O\_ID} uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



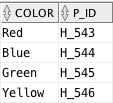
|  |  |  |
| --- | --- | --- |
| **Dependent\_Orders** |  |  |
| **Functional Dependencies** | {PSU\_ID,O\_ID,Name}-> {PSU\_ID,O\_ID,Name} | |
| **Primary Key** | {PSU\_ID,O\_ID,Name} | {PSU\_ID, O\_ID,Name} uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



|  |  |  |
| --- | --- | --- |
| **Product** |  |  |
| **Functional Dependencies** | P\_ID -> {Price,Size\_p,Stock} | |
| **Primary Key** | P\_ID | P\_ID uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



|  |  |  |
| --- | --- | --- |
| **O\_Contains** |  |  |
| **Functional Dependencies** | {P\_ID,O\_ID} -> {P\_ID,O\_ID} | |
| **Primary Key** | {P\_ID,O\_ID} | {P\_ID,O\_ID} uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |



|  |  |  |
| --- | --- | --- |
| **Color** |  |  |
| **Functional Dependencies** | {P\_ID,color}->{PID\_color} | |
| **Primary Key** | {P\_ID,color} | {P\_ID,color} uniquely identifies any other attribute in this table |
|  | **Decision** | **Reason** |
| **First Normal Form** | Yes | Table has a primary key  All rows are unique  Table has atomic values for all attributes |
| **Second Normal Form** | Yes | Follows 1NF  There is no partial dependency |
| **Third Normal Form** | Yes | Follows 2NF  There is no transitive dependency |
| **Boyce Codd Normal Form** | Yes | Follows 3NF  All Non primary key attribute depend only on Primary key attribute |

**Select Queries**

|  |  |  |  |
| --- | --- | --- | --- |
| Query 1 |  |  |  |
| **English version** | Return the name of all the users living in Malvern | | |
| **SQL sentence** | SELECT F\_Name, L\_Name, City  FROM User\_PSU  WHERE City LIKE '%Malvern%' | | |
| **Example of returned rows (screen caption)** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Query 2 |  |  |  |
| **English version** | Return the email address of users | | |
| **SQL sentence** | SELECT F\_NAME,Email  FROM User\_PSU  RIGHT JOIN User\_PSU\_email ON User\_PSU.PSU\_ID = User\_PSU\_email.PSU\_ID; | | |
| **Example of returned rows (screen caption)** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Query 3 |  |  |  |
| **English version** | Return the name of users and his/her dependents along woth the relationship. | | |
| **SQL sentence** | SELECT a.F\_Name AS Name ,b.Name AS Dependent,b.Relationship  FROM User\_PSU a , Dependant b  WHERE a.PSU\_ID = b.PSU\_ID; | | |
| **Example of returned rows (screen caption)** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Query 4 |  |  |  |
| **English version** | Return the students who have a GPA greater than 3.25 | | |
| **SQL sentence** | SELECT a.F\_Name as Name, b.GPA  FROM User\_PSU a , Student b  WHERE (a.PSU\_ID = b.PSU\_ID) AND GPA > 3.25; | | |
| **Example of returned rows (screen caption)** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Query 5 |  |  |  |
| **English version** | Return the list of orders subjected to delivery current month | | |
| **SQL sentence** | SELECT O\_ID,pickup\_date  FROM order\_t  WHERE trunc(pickup\_date, 'MON') = trunc(sysdate, 'MON') | | |
| **Example of returned rows (screen caption)** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Query 6 |  |  |  |
| **English version** | Return the total number of items in the inventory | | |
| **SQL sentence** | SELECT SUM(stock) AS Inventory\_Size  FROM product | | |
| **Example of returned rows (screen caption)** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Query 7 |  |  |  |
| **English version** | Return the count of products based upon color in descending order | | |
| **SQL sentence** | SELECT Color, count(color)  FROM color  WHERE color is not null  GROUP BY color  ORDER BY count(color) Desc; | | |
| **Example of returned rows (screen caption)** |  | | |