## BOOK WEB SHOP

|  |
| --- |
| SOFTWARE SYSTEM DEVELOPMENT  YEAR 2 |
| Student No: W20044623  Student Name: DMITRIJ JAKOVLEV |

TABLE OF CONTENT

[INTRODUCTION 3](#_Toc417580642)

[SAMPLE DATA 4](#_Toc417580643)

[BUSINESS PROCESS DESCRIPTION 5](#_Toc417580644)

[STATEMENT OF TASKS 6](#_Toc417580645)

[LIST OF TABLES 7](#_Toc417580646)

[Super/Sub Type representation 8](#_Toc417580647)

[Table Specification 13](#_Toc417580648)

[List of Frequently used Queries 19](#_Toc417580649)

[Security 19](#_Toc417580650)

[PART TWO 20](#_Toc417580651)

[CREATE TABLES 21](#_Toc417580652)

[INSERT DATA 29](#_Toc417580653)

[List of Frequently used queries and results of same 33](#_Toc417580654)

[Security 40](#_Toc417580655)

[CONCLUSIONS 41](#_Toc417580656)

# INTRODUCTION

As instructed in the brief for my database project I am required to investigate the data stored in a small business or organization and create an Oracle SQL database.

I am going to base my project on an online internet bookshop which consists from several tables, and technology where must be evaluating to make a better and faster decision. The aim of topics should include the management of data in Oracle SQL.

# SAMPLE DATA

* Customer (CustomerID, CName, CSurname, CEmail, CTelephone, City, Street, House, Flat)
* Orders (OrderID, CustomerID, Collection\_PountID, Order\_Date, Del\_Date, Payment Method)
* Book (BookID, Title, Genre, Price, Author, Publisher, Date of Release, Number of Pages)
* Order Line (OrderLineID, Total Price, Quantity, BookID, OrderID)
* Payment (PaymentID, OrderID, Payment Date, Amount, Today Date)
* CCheck (CheckNum, PaymentID, Bank Name)
* Card( CardNum, PaymentID, Card Type, Expire Date, Holder Name, Holder Address)
* Staff (StaffID, OrderID, STName, Email, Staff Job Title)
* Collection Point (Collection PointID, StaffID, Open Time)
* Customer Collection (OrderID)
* Delivery (DeliveryID, OrderID, Date of Delivery, Delivery Address, Delivery Charges, Status)
* Courier (CourierID, Name of Service)
* Transport (TransportID, Direction)

# BUSINESS PROCESS DESCRIPTION

Using Internet Book Shop customer log in on site where is situated assortment of production with additional information or detailed information. Every book has ID in catalogue, author, price, year of publishing, name of publisher. Also buyer has opportunities about type of payment, delivery type, contact information and details of administration.

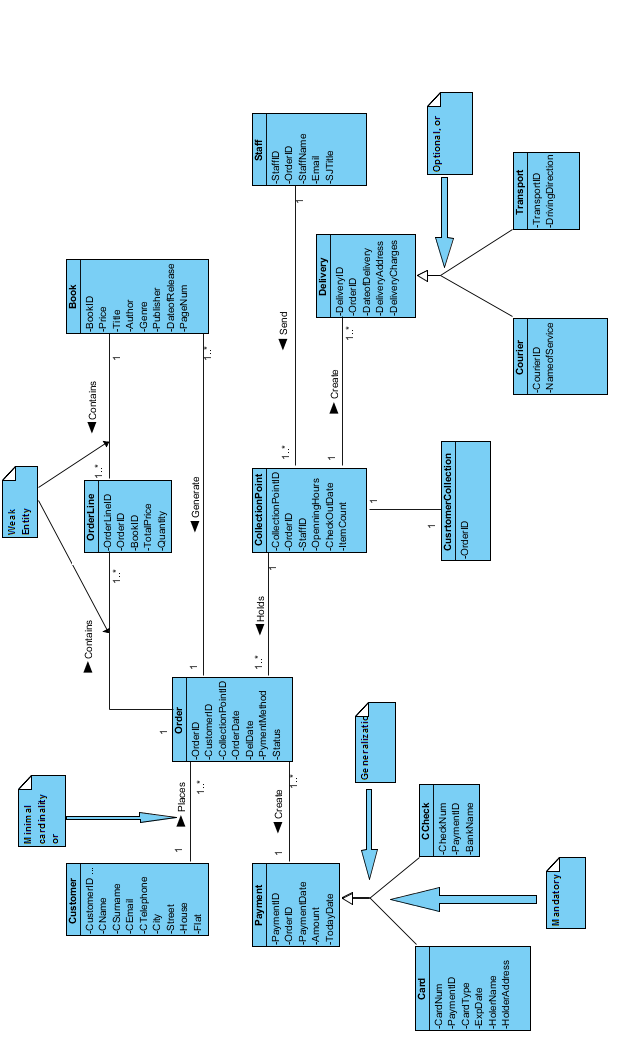
Customer to make an order firstly should make authorization and only after he will be able to log in on his personal page. After that customer looking for book which he wants to buy. He will able to use search engine and looking for by author, publisher and name of book. When he chose books he will able to proceed into the next step and apply the order.

For the purchase of books buyer indicates the delivery address, specifies the method of payment and sends the order to the manager of sales.

# STATEMENT OF TASKS

* The database should be stored list of books with the description of their properties, because the main purpose of an online store is selling them. .
* Certain employees responsible for each order. In order to keep track of it, as well as to the customers had the opportunity to made order by phone, clarify the status of an order or find out additional information about couriers and managers also must be stored in the. It is also necessary to store information about administrators and support staff, to managers or users of the service can solve technical problems.
* For the formation of the order and its execution in the database should be stored information about customers.
* In the case when customers choose payment method with credit card in the database can be stored card details.
* For the process of buying books and track order status in the database should be stored information about orders.
* For the distribution of books ordered over the collection points, as well as for the convenience of the client receives the order must be stored in the database a list of points of delivery of the goods.
* For the convenience of calculating the cost of delivery and for delivering information for customers and couriers should be stored information about each delivery.
* The part of purchase order that specifies the detailed information about a requested item should be stored in order line table for sake of space (weak entity).

# LIST OF TABLES



# Super/Sub Type representation

Payment can be made by credit card or by check on delivery or in collection point.

* All data can be represent in one table
* In number of subclasses
* In superclass and subclasses

One Table

|  |  |  |
| --- | --- | --- |
| Payment | Field Size | Description |
| PAYMENT ID | 25 | Indicator for each payment |
| ORDER ID | 25 | Indicator for each order |
| PAYMENT DATE | 8 | Date of payment |
| AMOUNT | 5 | Total amount of payment (excl. delivery charges) |
| TODAY DATE | 8 | Current day |

Total: 71 bytes

|  |  |  |
| --- | --- | --- |
| Payment | Field Size | Description |

|  |  |  |
| --- | --- | --- |
| CARD NUM | 14 | INDICATOR FOR EACH CARD |
| CARD TYPE | 25 | TYPE OF CARD |
| EXP DATE | 8 | CARD DATE EXPIRE |
| HOLDER NAME | 15 | NAME OF CARD HOLDER |
| HOLDER ADDRESS | 50 | ADDRESS OF CARD HODER |

Total: 112 bytes

|  |  |  |
| --- | --- | --- |
| PAYMENT | FIELD SIZE | DESCRIPTION |
| CHECK NUM | 10 | INDICATOR FOR EACH CARD |
| BANK NAME | 25 | NAME OF BANK |

Total: 35 bytes

### Option 1 one entity called payment

We put all data into one super table.

71+112+35 = 218 bytes required per record.

### 100 different payment thus 100 records = 21,800 bytes =21. 8 Kb

### Total for option 1 >>>> 21, 8 Kb

### Option 2 two tables one for card and one for check payment

Card table will consist of 71+ 112= 183 bytes for each record

80% of 100 payments are card payments = 80 payments

### Total storage for card payments is: 80 \* 112 = 8, 960 bytes = 8. 9 Kb

Check table will consist of 71+ 35 = 106 bytes for each record

20% Check payments = 20 payments

Total storage for a SKI vacations is 20 \* 106 = 2120 = 2.1 Kb

### Total for option 2 >>>> 8.9 + 2.1 = 11 Kb

Option 3 the entities one general data, one table for Card payment data and one table for Check payment data.

So, the general data required 71 bytes and for 100 records it will be 71 bytes \* 100 bytes =

7100 bytes =7, 1Kb

For table Card required 112 byte + primary key from payment table + order id where required additional 50 bytes = 112+50 = 162 bytes and 80% of card payments 162\*80 = 12,960 bytes = 12,96Kb

For table Check we need 35 bytes and additional space for payment id , and order id = 35+50 = 85 bytes where 20% of all payment made by check = 85\*20=1700 bytes = 1,7Kb

Total for option 3 >>>> 7, 1 + 12, 96 + 1, 7 = 21, 76Kb

The less space required for storage data is option 2 but if it depends on how customer wants to apply the queries. If for whole data the best option1 if query for methods of payments the best option 2 and three.

Another representation is delivery table which may or not split by two options. One is method of delivery by courier and another method of delivery by transports which particularly for a heavy orders and longer distance.

Count the storage space for delivery table on the same way like above.

One Table

|  |  |  |
| --- | --- | --- |
| PAYMENT | FIELD SIZE | DESCRIPTION |
| DELIVERY ID | 25 | INDICATOR FOR EACH DELIVERY |
| ORDER ID | 25 | INDICATOR FOR EACH ORDER |
| DELIVERY DATE | 8 | DATE WHEN WILL BE DELIVERY |
| DELIVERY ADDRESS | 50 | ADDRESS OF DESTINATION OF ORDER |
| DELIVERY CHARGES | 5 | CHARGES APPLY |

Total 113 bytes

|  |  |  |
| --- | --- | --- |
| PAYMENT | FIELD SIZE | DESCRIPTION |
| COURER ID | 15 | INDICATOR FOR EACH DELIVERY |
| SERVICE NAME | 25 | NAME OF COMPANY |

Total 40 bytes

|  |  |  |
| --- | --- | --- |
| PAYMENT | FIELD SIZE | DESCRIPTION |
| TRANSPORT ID | 10 | INDICATOR FOR EACH DELIVERY |
| DRIVING DIRECTION | 100 | WAY TO DESTINATION |

Total 110 bytes

We put all data into one super table.

113+40+110 = 263 bytes required per record.

### 100 records required = 26,300 bytes =26. 3 Kb

### Total for option 1 >>>> 26, 3 Kb

### Option 2 two tables one for courier and one for transport delivery

Courier table will consist of 113+40= 153 bytes for each record

90% of deliveries by courier = 90 deliveries

### Total storage for deliveries by courier is: 90\*153 = 13, 770 bytes = 13, 77Kb

Transport table will consist of 110+113 = 223 bytes for each record

10% of deliveries by = 10 deliveries

Total storage for a Transport table is 10 \*223 = 223 bytes = 22, 3Kb

### Total for option 2 >>>> 13, 77 + 22, 3 = 36, 07Kb

Option 3 the entities one general data, one table for Courier data and one table for Transport data.

So, the general data required 113 bytes and for 100 records it will be 113 bytes \* 100 bytes = 11300 bytes = 11, 3Kb

For table Courier required 40 byte + primary key from delivery table + order id where required additional 50 bytes = 90 bytes = 90 bytes and 90% by courier 90\*90 = 8,100 bytes = 8, 1Kb

For table Transport we need 110 bytes and additional space for payment id, and order id = 110+50 = 160 bytes where 10% of all deliveries by transport = 160\*10=1600 bytes = 1,6Kb

Total for option 3 >>>> 11, 3 + 8, 1+ 1, 3 = 20, 7Kb

For most queries the best option 1 and take the optimal space from another three.

# Table Specification

CUSTOMER

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Customer ID | Uniquely identifies a Customer ID | CHAR (10) PK | Not Null or Empty, Unique |
| CName | FName of customer | VARCHAR (25), | Not Null or Empty |
| CSurname | LName of customer | VARCHAR (25), | Not Null or Empty |
| CEmail | Customer email address | VARCHAR (23) | Must be of type assigned to the purchasing company... |
| CTelephone | For contacts | VARCHAR (10) | 10 digits only Check |
| City | Address for delivery an order | CHAR (25) | Not Null or Empty |
| Street | Address for delivery an order | VARCHAR (25), | Not Null or Empty |
| House | Address for delivery an order | VARCHAR (25) | Not Null or Empty |
| Flat | Address for delivery an order | VARCHAR (3), | Not Null or Empty |

ORDERS

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| OrderID | Uniquely IDENTIFIES ORDER ID | CHAR (25) PK | Not Null or Empty Check |
| CustomerID | Uniquely identifies a Customer ID | CHAR (10) FK | Not Null or Empty, Check |
| COLLECTION\_POINTID | Uniquely identifies a Customer ID | CHAR (10) FK | Not Null or Empty, Check |
| Order Date | Date when customer apply order | DATE | Not Null or Empty, Must no early than today DEFAULT SYSDATE |
| DelDate | Date of delivery | DATE | Delivery must be made <= 10 days, Check |
| Payment Method | Type of payment options for customer | CHAR (8) | Not Null or Empty |
| Status | Order accepted or not | char (1) | Default 0 |

BOOK

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| BookID | Uniquely IDENTIFIES BOOK ID | VARCHAR (20) PK | Not Null or Empty, Check |
| Price | The price of order | CHAR (5) | Not Null or Empty and price>0, positive |
| Title | The title of book | VARCHAR (100) | Not Null or Empty |
| Author | Name and Surname of writer | VARCHAR (50) | Not Null or Empty |
| Genre | Type of GENRE | VARCHAR (25) | Not Null or Empty |
| Publisher | Who publish that BOOK; hold all publishing info. | VARCHAR (25) | Null |
| Date of release | When book was appear in the sale | Date | Null |
| Page Numbers | Shows how many pages in the book | CHAR (5) | Null |

ORDER LINE

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| OrderLineID | Uniquely identifies an Order Line ID | CHAR (25) PK | Not Null or Empty, Auto increment, Check, Unique |
| OrderID | Uniquely IDENTIFIES ORDER Number | VARCHAR (25) FK | Not Null or Empty, Check |
| BookID | Each book has ID to easy search by that parameter. | VARCHAR (20) FK | Not Null or Empty, Check |
| Total Price | Total price for order LINE | CHAR (5) | Not Null or Empty, price>0, positive |
| Quantity | Quantity of ordered items | CHAR (5) | Not Null or Empty |

PAYMENT

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Payment ID | Uniquely identifies Payment Number | VARCHAR (25) PK | Not Null or Empty,Check, Unique, Auto increment |
| Order ID | Uniquely identifies Order Number | VARCHAR (25) FK | Not Null or Empty, Check |
| Payment Date | Date when payment should be made | Date | Not Null, Check |
| Today Date | Current date | Date | Default, Sysdate |
| Amount | Total amount of payment | CHAR (5) | Not Null, price>0, positive, Check |

CHECK

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| PAYMENT ID | UNIQUELY IDENTIFIES PAYMENT NUMBER | VARCHAR (25) PK | NOT NULL OR EMPTY, |
| Check Number | Unique number | CHAR (10) | Not null or Empty, Check, must be 10 digits |
| Bank Name | Name of Bank where assigned check | VARCHAR (25) | Not Null or Empty |

CARD

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| PAYMENT ID | UNIQUELY IDENTIFIES PAYMENT NUMBER | VARCHAR (25) PK | NOT NULL OR EMPTY, |
| Card Number | Unique number of card | CHAR (10) | no more than 14 digits. Check, Not Null or Empty |
| Card Type | Type of Card | VARCHAR (25) | Check, Not Null or EmptyNot Null or Empty |
| Expire Date | Date of card expire | DATE | Check, Not Null or Empty |
| Holder Name | Card holder name | VARCHAR (50) | Not Null or Empty |
| Holder Address | Address | VARCHAR (50) | Not Null or Empty |

COLLECTION POINT

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Collection Point ID | Unique Num of Collection Point | CHAR (10) PK | Unique, Not Null or Empty |
| Order ID | Unique Num of Order | Number (25) FK | Unique, Not Null or Empty |
| Staff ID | Unique Num of Staff | VARCHAR (25) | Unique, Not Null or Empty |
| Opening Hours | Time when point is open | CHAR (25) |  |
| Check out date | Date when order is ready for collection | CHAR (25) |  |
| Item count | Quantity of ordered items | Char (3) | Not Null |

COLLECT BY CUSTOMER

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Order ID | Unique Number of Order | VARCHAR (25)PK | Unique, Not Null or Empty |

STAFF

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Staff ID | Unique Num of Staff | CHAR (25) PK | Not Null or Empty, Unique, Check |
| Order ID | Unique Num of Order | VARCHAR (25) FK | Unique, Not Null or Empty |
| Staff Name | FName, LName of staff | VARCHAR (35), Alternate Key | Not Null or Empty |
| Email | Staff email for contact | VARCHAR(30) | Not Null |
| SJTITLE | Job title | VARCHAR (15) | Not Null |

DELIVERY

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Delivery ID | Unique Number of Delivery | Char (25) PK | Not Null or Empty, Unique, Check |
| Order ID | Unique Number of Order | Char (25) FK | Unique, Not Null or Empty |
| Date of Delivery | Assigned date of delivery | Date | Not Null |
| Delivery Address | Address of customer, Alternate Key | VARCHAR (50) | Not Null or Empty |
| Delivery Charges | Charges for delivery | CHAR (5) | Not Null or Empty and charges > 0, positive |

COURIER

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Courier ID | Unique Number of courier | Char (10) PK | Not Null or Empty, Unique, Check |
| Name of service | Delivery company name | VARCHAR (25) | Not Null or Empty |

TRANSPORT

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Transport ID | Unique Number of transport | Char (10) PK | Not Null or Empty, Unique, Check |
| Driving direction | Direction from collection point to receiver | VARCHAR(100) | Not Null or Empty |

# List of Frequently used Queries

* List all Programming books with price less than 20
* Find the book with lowest price
* Find orders payment was made by Card and already paid
* Retrieve all Customers Names, Payment Method and Order ID
* Retrieve all Customers Names, from Dublin
* Sort all staff by Job Title on ASC Order
* To look on data from two tables by creating view
* Retrieve Information About Payments, Customer Names, And Status, And Delivery Date
* List all Orders for Programming books costing under €25 where the publisher is Head First and STATUS = Y.
* Viewing records from a book without knowing exact details
* Select customers who create an order

.

.

.

.

# Security

BOOKS

To access this database have managers, administrators. Edit this section may exercise managers and administrators.

STAFF

Edit this section may exercise the administrators and staff.

PAYMENT

To access in this database have accountant. Edit this section may exercise by the accountant.

ORDERS

To access in this database have managers, administrators, couriers. Edit this section may exercise the administrators, managers.

DELIVERY

To access in this database have managers, administrators, couriers. Edit this section may exercise administrators and managers.

ORDER LINE ITEMS

To access in this database have administrator, staff, managers. Edit this section may exercise administrators and managers, and staff.

COLLECTION POINT

To access in this database have administrator, staff, and managers. Edit this section may exercise administrators and managers, staff.

COURIER AND TRANSPORT

To access in this database have administrator, managers, and couriers. Edit this section may exercise administrators, managers.

# 

# PART TWO

In part one of these projects I wrote a database design specification for WE BOOK SHOP. This part will contain the code segments required to create all tables and views needed for the database system. I will also supply code used to populate my oracle tables and set-up the access privileges to this data. In the final section I include the code needed to execute the queries specified in part 1 of this project.

# CREATE TABLES

CUSTOMER

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Customer ID | Uniquely identifies a Customer ID | CHAR (10) PK | Not Null or Empty, Unique |
| CName | FName of customer | VARCHAR (25), | Not Null or Empty |
| CSurname | LName of customer | VARCHAR (25), | Not Null or Empty |
| CEmail | Customer email address | VARCHAR (23) | Must be of type assigned to the purchasing company... |
| CTelephone | For contacts | VARCHAR (10) | 10 digits only Check |
| City | Address for delivery an order | CHAR (25) | Not Null or Empty |
| Street | Address for delivery an order | VARCHAR (25), | Not Null or Empty |
| House | Address for delivery an order | VARCHAR (25) | Not Null or Empty |
| Flat | Address for delivery an order | VARCHAR (3), | Not Null or Empty |

CREATE TABLE CUSTOMER (

CUSTOMERID CHAR(10)NOT NULL,

CNAME VARCHAR(25)NOT NULL,

CSURNAME VARCHAR(25)NOT NULL,

CEMAIL VARCHAR(22)NOT NULL,

CTELEPHONE VARCHAR(10)NOT NULL,

CITY CHAR(25)NOT NULL,

STREET VARCHAR(25)NOT NULL,

HOUSE VARCHAR(25)NOT NULL,

FLAT VARCHAR(3)NOT NULL,

CONSTRAINT CUSTOMER\_PK PRIMARY KEY(CUSTOMERID),

CONSTRAINT CTELEPHONE\_CK CHECK(LENGTH(CTELEPHONE)=10));

ORDERS

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| OrderID | Uniquely IDENTIFIES ORDER ID | CHAR (25) PK | Not Null or Empty Check |
| CustomerID | Uniquely identifies a Customer ID | CHAR (10) FK | Not Null or Empty, Check |
| COLLECTION\_POINTID | Uniquely identifies a Customer ID | CHAR (10) FK | Not Null or Empty, Check |
| Order Date | Date when customer apply order | DATE | Not Null or Empty, Must no early than today DEFAULT SYSDATE |
| DelDate | Date of delivery | DATE | Delivery must be made <= 10 days, Check |
| Payment Method | Type of payment options for customer | CHAR (8) | Not Null or Empty |
| Status | Order accepted or not | char (1) | Default 0 |

CREATE TABLE ORDERS(

ORDERID CHAR(25) NOT NULL,

CUSTOMERID CHAR(10)NOT NULL,

COLLECTION\_POINTID CHAR (10) NOT NULL,

ORDER\_DATE DATE DEFAULT SYSDATE NOT NULL,

DEL\_DATE DATE NOT NULL,

PAYMENT\_METH CHAR(8) NOT NULL,

STATUS CHAR(1) NOT NULL,

CONSTRAINT ORDERID\_PK PRIMARY KEY(ORDERID),

CONSTRAINT FK\_ORDER FOREIGN KEY(CUSTOMERID) REFERENCES CUSTOMER(CUSTOMERID)ON DELETE SET NULL,

CONSTRAINT DEL\_DATE\_CK CHECK(DEL\_DATE<=ORDER\_DATE+10));

BOOK

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| BookID | Uniquely IDENTIFIES BOOK ID | VARCHAR (20) PK | Not Null or Empty, Check |
| Price | The price of order | CHAR (5) | Not Null or Empty and price>0, positive |
| Title | The title of book | VARCHAR (100) | Not Null or Empty |
| Author | Name and Surname of writer | VARCHAR (50) | Not Null or Empty |
| Genre | Type of GENRE | VARCHAR (25) | Not Null or Empty |
| Publisher | Who publish that BOOK; hold all publishing info. | VARCHAR (25) | Null |
| Date of release | When book was appear in the sale | Date | Null |
| Page Numbers | Shows how many pages in the book | CHAR (5) | Null |

CREATE TABLE BOOK(

BOOKID VARCHAR(20)NOT NULL,

PRICE CHAR(5)NOT NULL,

TITLE VARCHAR(100)NOT NULL,

AUTHOR VARCHAR(50) NOT NULL,

GENRE VARCHAR(25)NOT NULL,

PUBLISHER VARCHAR(100)NOT NULL,

REL\_DATE DATE,

PAGE\_NUM CHAR(5)NOT NULL,

CONSTRAINT BOOKID\_PK PRIMARY KEY(BOOKID),

CONSTRAINT PRICE\_CK CHECK(PRICE>0));

ORDERLINE

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| OrderLineID | Uniquely identifies an Order Line ID | CHAR (25) PK | Not Null or Empty, Auto increment, Check, Unique |
| OrderID | Uniquely IDENTIFIES ORDER Number | VARCHAR (25) FK | Not Null or Empty, Check |
| BookID | Each book has ID to easy search by that parameter. | VARCHAR (20) FK | Not Null or Empty, Check |
| Total Price | Total price for order LINE | CHAR (5) | Not Null or Empty, price>0, positive |
| Quantity | Quantity of ordered items | CHAR (5) | Not Null or Empty |

CREATE TABLE ORDERLINE(

ORDERLINEID CHAR(25)NOT NULL,

ORDERID CHAR(25)NOT NULL,

BOOKID VARCHAR(20)NOT NULL,

QUANTITY CHAR(5)NOT NULL,

TOTALPRICE CHAR(5) NOT NULL,

CONSTRAINT ORDERLINE\_PK PRIMARY KEY(ORDERLINEID),

CONSTRAINT FK\_ORDERLINE\_BOOK FOREIGN KEY(BOOKID) REFERENCES BOOK(BOOKID),

CONSTRAINT FK\_ORDERLINE FOREIGN KEY(ORDERID) REFERENCES ORDERS(ORDERID),

CONSTRAINT CHK\_TOTALPRICE CHECK(TOTALPRICE > 0)

);

**PAYMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Payment ID | Uniquely identifies Payment Number | VARCHAR (25) PK | Not Null or Empty,Check, Unique, Auto increment |
| Order ID | Uniquely identifies Order Number | VARCHAR (25) FK | Not Null or Empty, Check |
| Payment Date | Date when payment should be made | Date | Not Null, Check |
| Today Date | Current date | Date | Default, Sysdate |
| Amount | Total amount of payment | CHAR (5) | Not Null, price>0, positive, Check |

CREATE TABLE PAYMENT(

PAYMENTID VARCHAR(25)NOT NULL,

ORDERID CHAR(25)NOT NULL,

PAYMENTDATE DATE NOT NULL,

AMOUNT CHAR(5)NOT NULL,

TODAY\_DATE DATE DEFAULT SYSDATE,

CONSTRAINT PAYMENT\_PKEY PRIMARY KEY (PAYMENTID),

CONSTRAINT FK\_PAYMENT FOREIGN KEY(ORDERID) REFERENCES ORDERS(ORDERID),

CONSTRAINT CHK\_PAYMENTDATE CHECK(PAYMENTDATE<=TODAY\_DATE),

CONSTRAINT CHK\_AMOUNT CHECK(AMOUNT > 0));

CCHECK

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| PAYMENT ID | UNIQUELY IDENTIFIES PAYMENT NUMBER | VARCHAR (25) PK | NOT NULL OR EMPTY, |
| Check Number | Unique number | CHAR (10) | Not null or Empty, Check, must be 10 digits |
| Bank Name | Name of Bank where assigned check | VARCHAR (25) | Not Null or Empty |

CREATE TABLE CCHECK(

CHECKNUM CHAR(10) NOT NULL,

PAYMENTID VARCHAR(25)NOT NULL,

BANKNAME VARCHAR(25) NOT NULL,

CONSTRAINT CCHECK\_PKEY PRIMARY KEY (CHECKNUM),

CONSTRAINT FK\_CCHECK FOREIGN KEY(PAYMENTID) REFERENCES PAYMENT(PAYMENTID),

CONSTRAINT CHECKNUM\_CK CHECK(LENGTH(CHECKNUM)=10));

CARD

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| PAYMENT ID | UNIQUELY IDENTIFIES PAYMENT NUMBER | VARCHAR (25) PK | NOT NULL OR EMPTY, |
| Card Number | Unique number of card | CHAR (10) | no more than 14 digits. Check, Not Null or Empty |
| Card Type | Type of Card | VARCHAR (25) | Check, Not Null or EmptyNot Null or Empty |
| Expire Date | Date of card expire | DATE | Check, Not Null or Empty |
| Holder Name | Card holder name | VARCHAR (50) | Not Null or Empty |
| Holder Address | Address | VARCHAR (50) | Not Null or Empty |

CREATE TABLE CARD(

CARDNUM CHAR(10) NOT NULL,

PAYMENTID VARCHAR(25)NOT NULL,

CARDTYPE VARCHAR(25) NOT NULL,

EXPDATE DATE NOT NULL,

HOLDERNAME VARCHAR(15)NOT NULL,

HOLDERADDR VARCHAR(50)NOT NULL,

CONSTRAINT CARD\_PK PRIMARY KEY(CARDNUM),

CONSTRAINT FK\_CARD FOREIGN KEY(PAYMENTID) REFERENCES PAYMENT(PAYMENTID),

CONSTRAINT CARDNUM\_CK CHECK(LENGTH(CARDNUM)=10));

STAFF

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Staff ID | Unique Num of Staff | CHAR (25) PK | Not Null or Empty, Unique, Check |
| Order ID | Unique Num of Order | VARCHAR (25) FK | Unique, Not Null or Empty |
| Staff Name | FName, LName of staff | VARCHAR (35), Alternate Key | Not Null or Empty |
| Email | Staff email for contact | VARCHAR(30) | Not Null |
| Sjtitle | Job title | VARCHAR (15) | Not Null |

CREATE TABLE STAFF(

STAFFID CHAR(25)NOT NULL,

ORDERID CHAR(25) NOT NULL,

STNAME VARCHAR(35) NOT NULL,

EMAIL VARCHAR(30) NOT NULL,

SJTITLE VARCHAR(15)NOT NULL,

CONSTRAINT STAFF\_PK PRIMARY KEY(STAFFID),

CONSTRAINT FK\_STAFF FOREIGN KEY(ORDERID) REFERENCES ORDERS(ORDERID));

COLLECTIONPOINT

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Collection Point ID | Unique Num of Collection Point | CHAR (10) PK | Unique, Not Null or Empty |
| Order ID | Unique Num of Order | Number (25) FK | Unique, Not Null or Empty |
| Staff ID | Unique Num of Staff | VARCHAR (25) | Unique, Not Null or Empty |
| Opening Hours | Time when point is open | CHAR (25) |  |
| Check out date | Date when order is ready for collection | CHAR (25) |  |

CREATE TABLE COLLECTIONPOINT(

COLLECTION\_POINTID CHAR (10)NOT NULL,

STAFFID CHAR (25)NOT NULL,

OPNTM CHAR (25)//

CHKOUTTIME CHAR (25),

CONSTRAINT COLLECTIONPOINT\_PK PRIMARY KEY (COLLECTION\_POINTID),

CONSTRAINT FK\_COLLECTIONPOINT FOREIGN KEY (STAFFID) REFERENCES STAFF(STAFFID));

CUSTCOLL

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Order ID | Unique Number of Order | VARCHAR (25)PK | Unique, Not Null or Empty |

CREATE TABLE CUSTCOLL (

ORDERID CHAR (25) NOT NULL,

CONSTRAINT FK\_CUSTCOLL FOREIGN KEY(ORDERID) REFERENCES ORDERS(ORDERID));

DELIVERY

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Delivery ID | Unique Number of Delivery | Char (25) PK | Not Null or Empty, Unique, Check |
| Order ID | Unique Number of Order | Char (25) FK | Unique, Not Null or Empty |
| Date of Delivery | Assigned date of delivery | Date | Not Null |
| Delivery Address | Address of customer, Alternate Key | VARCHAR (50) | Not Null or Empty |
| Delivery Charges | Charges for delivery | CHAR (5) | Not Null or Empty and charges > 0, positive |

CREATE TABLE DELIVERY(

DELID CHAR (25) NOT NULL,

ORDERID CHAR (25) NOT NULL,

DL\_DATE DATE NOT NULL,

DEL\_ADDRSS VARCHAR (50)NOT NULL,

DCHARGES CHAR (5)NOT NULL,

CONSTRAINT DELIVERY\_PK PRIMARY KEY (DELID),

CONSTRAINT FK\_DELIVERY FOREIGN KEY (ORDERID) REFERENCES ORDERS(ORDERID),

CONSTRAINT CHK\_DCHARGES CHECK (DCHARGES > 0));

COURIER

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Courier ID | Unique Number of courier | Char (10) PK | Not Null or Empty, Unique, Check |
| Name of service | Delivery company name | VARCHAR (25) | Not Null or Empty |

CREATE TABLE COURIER (

COURIERID CHAR (15) NOT NULL,

SERVNAME VARCHAR (25) NOT NULL,

CONSTRAINT COURIER\_PK PRIMARY KEY(COURIERID));

TRANSPORT

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Description | Type | Constraint |
| Transport ID | Unique Number of transport | Char (10) PK | Not Null or Empty, Unique, Check |
| Driving direction | Direction from collection point to receiver | VARCHAR(100) | Not Null or Empty |

CREATE TABLE TRANSPORT (

TRANSPORTID CHAR (10) NOT NULL,

DRIVDIRECT VARCHAR (100) NOT NULL,

CONSTRAINT TRANSPORT\_PK PRIMARY KEY (TRANSPORTID));

# INSERT DATA

INSERT INTO CUSTOMER

INSERT INTO CUSTOMER VALUES(1000, 'Full', 'Priv', 'boab@hotmail.com', 9999999999,'New York', '5th Avenu', '1 St', 100);

INSERT INTO CUSTOMER VALUES(1234567893, 'Alex', 'Petroff', 'g@mail.com', 01857419623,'Dublin', 'Anne Lane', '43 Down', 43);

INSERT INTO CUSTOMER VALUES(200, 'Alex', 'Petroff', 'g@mail.com', 01857419623,'Dublin', 'Anne Lane', '43 Down', 43);

INSERT INTO CUSTOMER VALUES(201, 'Alex', 'Petroff', 'g@mail.com', 01857419623,'Dublin', 'Anne Lane', '43 Down', 43);

INSERT INTO CUSTOMER VALUES(202, 'Alex', 'Petroff', 'g@mail.com', 01857419623,'Dublin', 'Anne Lane', '43 Down', 43);

INSERT INTO CUSTOMER VALUES(203, 'Alex', 'Petroff', 'g@mail.com', 01857419623,'Dublin', 'Anne Lane', '43 Down', 43);

INSERT INTO CUSTOMER VALUES(205, 'John', 'Ivanoff', 'g@mail.com', 01857419587,'Longford', 'Anne Lane', '43 Down', 03

INSERT INTO CUSTOMER VALUES(207, 'Fill', 'Simirnoff', 'j@mail.com', 01858419623,'Limereck', 'Anne Lane', '43 Down', 5);

INSERT INTO CUSTOMER VALUES(208, 'Kst', 'Prutkoff', 'k@mail.com', 01857492123,'Kilkenny', 'Anne Lane', '43 Down', 6);

INSERT INTO CUSTOMER VALUES(209, 'Boris', 'Lery', 'ag@hotmail.com', 7414785987,'Galway', 'Bris St', '56/21', 68);

INSERT INTO CUSTOMER VALUES(210, 'Katty', 'Samuel', 'bt@rambler.eu', 3658745895,'Sligo', 'Andre Avenu', '65-12', 5);

INSERT INTO CUSTOMER VALUES(211, 'Fill', 'Abtaham', 'gf@pop.com', 6987458965,'Ballymurphy', 'Thomas St', '98-65 Down', 3);

INSERT INTO CUSTOMER VALUES(212, 'Kalif', 'Nasredinnovich', 'jy@etto.lt', 3652147527,'Buttevant', 'Taikos Lane', 'Saint Well', 7);

INSERT INTO CUSTOMER VALUES(213, 'Bobius', 'Shekspirus', 'kmail@mail.ru', 3658974521,'Hollyford', 'Partizany Street', 'Blump High', 9);

INSERT INTO CUSTOMER VALUES(214, 'Celovekus', 'Vulgarisus', 'st@gums.ie', 2569874521,'Portroe', 'Barrack Down', 'Under Low', 2);

INSERT INTO CUSTOMER VALUES(215, 'Edgard', 'Po', 'we@fgt.com', 3658974521,'Waterville', 'Lower Bay', '68 efst', 54);

INSERT INTO BOOK

INSERT INTO book VALUES(1, 25, 'Java Developer', 'Peter Trost', 'Programming', 'Head First', '02-May-2005', 200);

INSERT INTO book VALUES(2, 25, 'Java Developer', 'Peter Trost', 'Programming', 'Head First', '02-May-2005', 200);

INSERT INTO book VALUES(3, 25, 'Java Developer', 'Peter Trost', 'Programming', 'Head First', '02-May-2005', 200);

INSERT INTO book VALUES(4, 25, 'Java Developer', 'Peter Trost', 'Programming', 'Head First', '02-May-2005', 200);

INSERT INTO book VALUES(6, 15, 'Android', 'Trust', 'Programming', 'Head First', '02-May-2005', 200);

INSERT INTO book VALUES(7, 35, 'HTML', 'Kust', 'Programming', 'Head First', '02-May-2005', 200);

INSERT INTO book VALUES(8, 20, 'SQL PLUS', 'Zmust', 'Programming', 'Head First', '02-May-2005', 200);

INSERT INTO book VALUES(10, 12, 'MySQL', 'Prust', 'Programming', 'Head First', '02-May-2005', 200);

INSERT INTO book VALUES(11, 55, 'Second World War', 'Stalin I.V.', 'History', 'RussBalt', '02-May-2011', 355);

INSERT INTO book VALUES(12, 32, 'Pirr', 'Alan Din Foster', 'Fantasy', 'RegMix', '02-May-2013', 355);

INSERT INTO book VALUES(13, 31, 'Android - Tech View', 'Vasilij Koshkin', 'Fantasy', 'TView', '01-Jan-2014', 235);

INSERT INTO book VALUES(14, 65, 'Science of Code', 'Michael Vov', 'Technology', 'ModernStyle', '25-Dec-2015', 155);

INSERT INTO book VALUES(15, 16, 'The Ruling Class', 'Caetano Mosca', 'Politics', ' McGrawHillCompany', '23-Jun-2008', 368);

INSERT INTO book VALUES(16, 68, 'Scientific Industries', 'Alexander Watt', 'Technology', 'Science', '06-Apr-1998', 297);

INSERT INTO ORDERS

INSERT INTO ORDERS VALUES (001, 1234567893, '08-Apr-2015', '09-Apr-2015', 'Card', 'Y');

INSERT INTO ORDERS VALUES (002, 1234567893, '08-Apr-2015', '09-Apr-2015', 'Card', 'Y');

INSERT INTO ORDERS VALUES (003, 1234567893, '08-Apr-2015', '09-Apr-2015', 'Card', 'Y');

INSERT INTO ORDERS VALUES (004, 1234567893, '08-Apr-2015', '09-Apr-2015', 'Card', 'Y');

INSERT INTO ORDERS VALUES (005, 1234567893, '08-Apr-2015', '09-Apr-2015', 'Card', 'Y');

INSERT INTO ORDERS VALUES (006, 201, '08-Jan-2015', '09-Apr-2015', 'Cash', 'Y');

INSERT INTO ORDERS VALUES (007, 202, '08-Dec-2015', '09-Apr-2015', 'Check', 'N');

INSERT INTO ORDERS VALUES (008, 204, '08-Feb-2015', '09-Apr-2015', 'Cash', 'Y');

INSERT INTO ORDERS VALUES (009, 210, '12-Mar-2013', '19-Mar-2013', 'Card', 'N');

INSERT INTO ORDERS VALUES (010, 211, '24-Feb-2015', '09-Mar-2015', 'Check', 'N');

INSERT INTO ORDERS VALUES (011, 212, '08-Apr-2012', '15-Apr-2012', 'Check', 'Y');

INSERT INTO ORDERS VALUES (012, 213, '25-Jul-2011', '03-Jul-2011', 'Card', 'N');

INSERT INTO ORDERS VALUES (013, 214, '12-Aug-2015', '19-Aug-2015', 'Cash', 'Y');

INSERT INTO ORDERS VALUES (014, 215, '21-Nov-2015', '30-Nov-2015', 'Check', 'Y');

INSERT INTO ORDERS VALUES (015, 209, '06-Oct-2009', '09-Oct-2009', 'Cash', 'N');

INSERT INTO ORDERLINE

INSERT INTO ORDERLINE VALUES (100, 001, 1, 50, 2);

INSERT INTO ORDERLINE VALUES (101, 002, 1, 50, 2);

INSERT INTO ORDERLINE VALUES (103, 003, 1, 50, 2);

INSERT INTO ORDERLINE VALUES (104, 004, 2, 50, 2);

INSERT INTO ORDERLINE VALUES (105, 005, 3, 32, 2);

INSERT INTO ORDERLINE VALUES (107, 00, 4, 52, 2);

INSERT INTO ORDERLINE VALUES (106, 5, 13, 3, 93);

INSERT INTO ORDERLINE VALUES (107, 7, 14, 5, 325);

INSERT INTO ORDERLINE VALUES (108, 9, 15, 7,112 );

INSERT INTO ORDERLINE VALUES (109, 10, 16, 1, 68);

INSERT INTO ORDERLINE VALUES (110, 11, 11, 8, 440);

INSERT INTO ORDERLINE VALUES (111, 12, 12, 9, 288);

INSERT INTO ORDERLINE VALUES (112, 13, 8, 4, 80);

INSERT INTO PAYMENT

INSERT INTO PAYMENT VALUES(1, 2, '8-Apr-2015', 50, '08-Apr-2015')

INSERT INTO PAYMENT VALUES(2, 1, '8-Apr-2015', 50, '08-Apr-2015');

INSERT INTO PAYMENT VALUES(3, 2, '8-Apr-2015', 50, '08-Apr-2015');

INSERT INTO PAYMENT VALUES(4, 3, '8-Apr-2015', 50, '08-Apr-2015');

INSERT INTO PAYMENT VALUES(8, 12, '01-Apr-11', 250, '23-Apr-2015');

INSERT INTO PAYMENT VALUES(9, 13, '12-Feb-15', 125, '23-Apr-2015');

INSERT INTO PAYMENT VALUES(10, 14, '21-Mar-15', 20, '23-Apr-2015');

INSERT INTO PAYMENT VALUES(11, 15, '06-Oct-09', 235, '23-Apr-2015');

INSERT INTO PAYMENT VALUES(12, 10, '08-Jan-15', 152, '23-Apr-2015');

INSERT INTO CCHECK

INSERT INTO CCHECK VALUES (9876543213, 1, ‘AIB’);

INSERT INTO CCHECK VALUES(9876543212,2, 'Ulster');

INSERT INTO CCHECK VALUES(9876543211,3, 'Bank of Ireland');

INSERT INTO CCHECK VALUES(9876543210,4, 'PRTB');

INSERT INTO CCHECK VALUES(8574165897,11, 'B&D Bank');

INSERT INTO CCHECK VALUES(3658741256,7, 'Swed Bank');

INSERT INTO CCHECK VALUES(8574165897,12, 'Parysh Credit');

INSERT INTO CARD

INSERT INTO CARD VALUES(5987458799, 1, 'Debit', '02-May-2016', 'John Brown', 'USA, NY, 5th Avenu');

INSERT INTO CARD VALUES(5987458798, 2, 'Credit', '25-Aug-2018', 'John Red', 'Canada, Ottava, 5th Avenu');

INSERT INTO CARD VALUES(5987458797, 3, 'MasterCard', '05-Sep-2020', 'John White', 'Ireland, Cork, 5TH Square');

INSERT INTO CARD VALUES(5987458796, 4, 'VisaGold', '18-Feb-2019', 'John Gray', 'UK, Liverpool, Island Street');

INSERT INTO CARD VALUES(6958741256, 7, 'Maestro Card', '14-May-2021', 'Alex Boldwin', 'USA, Boston, Avenue 14/21');

INSERT INTO CARD VALUES(5987455698, 8, 'Visa Platinum', '06-Jan-2016', 'Silvestr Stallone', 'ESP, Madrid, Saint Pat');

INSERT INTO CARD VALUES(5987452012, 9, 'SwedMedCard', '04-Apr-2017', 'John Rembo', 'RUS, Piter, Partizany - 22');

INSERT INTO CARD VALUES(5987450036, 10, 'AIB Student', '09-Dec-2015', 'Potc Valera', 'IRE, Cork, Mary Street');

INSERT INTO STAFF

INSERT INTO STAFF VALUES(1001, 001, 'Michail Gorbachev', 'cust@hotmail.com', 'Customer');

INSERT INTO STAFF VALUES(1000, 001, 'Barack Obama', 'boba@hotmail.com', 'Manager');

INSERT INTO STAFF VALUES(500, 001, 'Mors Tors', 'mt@mail.eu', 'Despatcher');

INSERT INTO STAFF VALUES(501, 002, 'Cus Tus', 'ct@hotmail.ie', 'Accounter');

INSERT INTO STAFF VALUES(502, 003, 'Shmus Plus', 'sp@gmail.com', 'Warehouse');

INSERT INTO STAFF VALUES(503, 004, 'Gnus Zmus', 'gz@wit.ie', 'Driver');

INSERT INTO STAFF VALUES(504, 004, 'Gnus Zmus', 'gz@wit.ie', 'Driver');

INSERT INTO STAFF VALUES(505, 003, 'Doddy Doddy', 'PL@mail.eu', 'Administrator');

INSERT INTO STAFF VALUES(509, 005, 'Smuli Smuli', 'TY@wit.ie', 'Dispatcher');

INSERT INTO STAFF VALUES(507, 007, 'Muli Muli', 'TR@gmail.com', 'Warehouseman');

INSERT INTO STAFF VALUES(511, 007, 'Fast as Bullet', 'fb@gmail.com', 'Courier');

INSERT INTO COLLECTION POINT

INSERT INTO COLLECTIONPOINT VALUES(255,500, '05-Aug-2015', '08-Feb-2015' );

INSERT INTO COLLECTIONPOINT VALUES(256,501, '05-Aug-2015', '08-Feb-2015' );

INSERT INTO COLLECTIONPOINT VALUES(257,502, '05-Aug-2015', '08-Feb-2015' );

INSERT INTO COLLECTIONPOINT VALUES(258,503, '05-Aug-2015', '08-Feb-2015' );

INSERT INTO COLLECTIONPOINT VALUES(001, 500, '05-Aug-2015','08-Feb-2015');

INSERT INTO COLLECTIONPOINT VALUES(002, 501, 'Mon-Fri,9:00-17:00','08-Feb-2015');

INSERT INTO CUSTCOLL

INSERT INTO CUSTCOLL VALUES(001);

INSERT INTO CUSTCOLL VALUES(001);

INSERT INTO CUSTCOLL VALUES(001);

INSERT INTO CUSTCOLL VALUES(001);

INSERT INTO CUSTCOLL VALUES(001);

INSERT INTO CUSTCOLL VALUES(001);

INSERT INTO CUSTCOLL VALUES(003);

INSERT INTO CUSTCOLL VALUES(002);

INSERT INTO CUSTCOLL VALUES(004);

INSERT INTO DELIVERY

INSERT INTO DELIVERY VALUES(10,001,'01-Aug-2015', 'Dublin, Low Street 13, App 4', 10);

INSERT INTO DELIVERY VALUES(11,002,'21-Aug-2015', 'Cork, Place Hall 01, Flat 10', 5);

INSERT INTO DELIVERY VALUES(12, 003, '01-Sep-2015', 'Waterford, Barrack Street 15', 10);

INSERT INTO DELIVERY VALUES(13, 004, '21-Jun-2015', 'Longford, Penrose Lane 15',5);

INSERT INTO DELIVERY VALUES(14, 005, '26-Jul-2015', 'Longford Harbour Point 55',15);

INSERT INTO DELIVERY VALUES(15,001,'01-Aug-2015', 'Sligo, Low Street 13, App 4', 10);

INSERT INTO DELIVERY VALUES(16,002,'21-Aug-2015', 'Athlone, Place Hall 01, Flat 10', 5);

INSERT INTO DELIVERY VALUES(17, 003, '01-Sep-2015', 'New Ross, Barrack Street 15', 10);

INSERT INTO DELIVERY VALUES(18, 004, '21-Jun-2015', 'Bray, Penrose Lane 15',5);

INSERT INTO DELIVERY VALUES(19, 005, '26-Jul-2015', 'Limerick Harbour Point 55',15);

INSERT INTO COURIER

INSERT INTO COURIER VALUES(201, 'Tdex');

INSERT INTO COURIER VALUES(202, 'ParcellServ');

INSERT INTO COURIER VALUES(203, 'FastDelivery');INSERT INTO COURIER VALUES(204, 'FeDex');

INSERT INTO COURIER VALUES(205, 'FastWay');

INSERT INTO COURIER VALUES(206, 'FlySpeed');

INSERT INTO COURIER VALUES(207, 'Bistro');

INSERT INTO COURIER VALUES(208, 'StarWa');

INSERT INTO TRANSPORT

INSERT INTO TRANSPORT VALUES(301, 'Stadium, Queens Road ON LEFT

GO ON Queens Road TOWARDS MG Road Junction,

CONTINUE INTO Lavelle Road');

INSERT INTO TRANSPORT VALUES(302, 'Black Road ON LEFT

GO ON White Road TOWARDS MG Road Junction,

CONTINUE INTO Lavelle Road');

INSERT INTO TRANSPORT VALUES(987, 'Start from Tesco, first traffic light, left, to M1-7 MILES,AFTER TOLL NEXT RIGHT-FIRST, RED HOUSE');

INSERT INTO TRANSPORT VALUES(303, 'Cork-Waterford-Kilkenny(Patrick Street, South House App 34, Tel 0857458956');

INSERT INTO TRANSPORT VALUES(304, 'Dublin-Athlone(M50)-Sligo(Castle Street - High Street - 7 OConnell Street 24');

# 

# List of Frequently used queries and results of same

List all Programming books with price less than 20

SELECT price, genre

FROM book

WHERE (Price<20) AND (Genre='Programming');

PRICE GENRE

----- -------------------------

15 Programming

12 Programming

2 rows selected.

Find the book with lowest price

Select BookID, Title, Price

FROM book

WHERE Price = (select MIN(Price) FROM book);

BOOKID TITLE PRICE

-------------------- ---------------------------------------------------------------------------------------------------- -----

10 MySQL 12

1 row selected.

Find orders payment was made by Card and already paid

SELECT ORDERID, CUSTOMERID, PAYMENT\_METH, STATUS

FROM ORDERS

WHERE PAYMENT\_METH = 'Card' AND Status = 'Y';

ORDERID CUSTOMERID PAYMENT\_ S

------------------------- ---------- -------- -

1 1234567893 Card Y

2 1234567893 Card Y

3 1234567893 Card Y

4 1234567893 Card Y

5 1234567893 Card Y

**5 rows selected.**

Retrieve all Customers Names, Payment Method and OrderID

SELECT ORDERS.ORDERID, CUSTOMER.CNAME, CUSTOMER.CSURNAME, ORDERS.PAYMENT\_METH  
FROM ORDERS  
INNER JOIN CUSTOMER  
ON ORDERS.CUSTOMERID=CUSTOMER.CUSTOMERID;

ORDERID CNAME CSURNAME PAYMENT\_METH

------------------------- ------------------------- ------------------------- --------

7 Alex Petroff Check

5 Alex Petroff Card

4 Alex Petroff Card

3 Alex Petroff Card

2 Alex Petroff Card

1 Alex Petroff Card

9 John Ivanoff Card

10 Bob Sidoroff Check

8 rows selected.

Retrieve all Customers Names, from Dublin

SELECT \* FROM CUSTOMER

WHERE

CNAME IN (SELECT CNAME FROM CUSTOMER WHERE CITY = 'Dublin') ;

CUSTOMERID CNAME CSURNAME CEMAIL CTELEPHONE CITY STREET HOUSE FLAT

---------- ------------------------- ------------------------- ---------------------- ---------- ------------------------- ------------------------- ------------------------- ---

1234567893 Alex Petroff g@mail.com 1857419623 Dublin Anne Lane 43 Down 43

1234567892 Alex Petroff g@mail.com 1857419623 Dublin Anne Lane 43 Down 43

1234567891 Alex Petroff g@mail.com 1857419623 Dublin Anne Lane 43 Down 43

1234567890 Alex Petroff g@mail.com 1857419623 Dublin Anne Lane 43 Down 43

204 Alex Petroff g@mail.com 1857419623 Dublin Anne Lane 43 Down 43

203 Alex Petroff g@mail.com 1857419623 Dublin Anne Lane 43 Down 43

202 Alex Petroff g@mail.com 1857419623 Dublin Anne Lane 43 Down 43

201 Alex Petroff g@mail.com 1857419623 Dublin Anne Lane 43 Down 43

200 Alex Petroff g@mail.com 1857419623 Dublin Anne Lane 43 Down 43

9 rows selected.

Sort all stuff by Job Title by ASC Order

SELECT \* FROM STAFF ORDER BY (CASE SJTITLE WHEN 'DRIVER' THEN 1

WHEN ‘ADMINISTRATOR’ THEN 2

WHEN 'MANAGER' THEN 3

WHEN 'DISPATCHER' THEN 4

ELSE 10 END) DESC, SJTITLE ASC;

STAFFID ORDERID STNAME EMAIL SJTITLE

------------------------- ------------------------- ----------------------------------- ------------------------------ ---------------

501 2 Cus Tus ct@hotmail.ie Accounter

505 3 Doddy Doddy PL@mail.eu Administrator

500 1 Mors Tors mt@mail.eu Despatcher

509 5 Smuli Smuli TY@wit.ie Dispatcher

504 4 Gnus Zmus gz@wit.ie Driver

503 4 Gnus Zmus gz@wit.ie Driver

502 3 Shmus Plus sp@gmail.com Warehouse

1. 7 Muli Muli TR@gmail.com Warehouseman
2. Rows selected.

To watch data from two tables by creating view

CREATE VIEW ORDERSVIEW

AS SELECT ORDERID, DEL\_DATE, CUSTOMER.CUSTOMERID,

CNAME, CITY

FROM ORDERS, CUSTOMER

WHERE CUSTOMER.CUSTOMERID=ORDERS.CUSTOMERID;

View created.

SELECT\* FROM ORDERSVIEW;

ORDERID DEL\_DATE CUSTOMERID CNAME CITY

------------------------- --------- ---------- ------------------------- -------------------------

7 09-APR-15 202 Alex Dublin

5 09-APR-15 1234567893 Alex Dublin

4 09-APR-15 1234567893 Alex Dublin

3 09-APR-15 1234567893 Alex Dublin

2 09-APR-15 1234567893 Alex Dublin

1 09-APR-15 1234567893 Alex Dublin

9 09-APR-15 205 John Longford

10 09-APR-15 206 Bob Cork

1. rows selected.

Retrieve information about payments, customer names, and status, and delivery date

SELECT ORDER\_DATE, CNAME, AMOUNT, STATUS

FROM ORDERS, CUSTOMER, PAYMENT

WHERE ORDERS.CUSTOMERID = CUSTOMER.CUSTOMERID;

ORDER\_DAT CNAME AMOUN STATUS

--------- ------------------------- ----- -

08-DEC-15 Alex 50 N

08-DEC-15 Alex 50 N

08-DEC-15 Alex 50 N

08-DEC-15 Alex 50 N

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-APR-15 Alex 50 Y

08-MAY-15 John 50 N

08-MAY-15 John 50 N

08-MAY-15 John 50 N

08-MAY-15 John 50 N

08-AUG-15 Bob 50 N

08-AUG-15 Bob 50 N

08-AUG-15 Bob 50 N

08-AUG-15 Bob 50 N

32 rows selected.

List all Orders for Programming books costing under €25 where the publisher is Head First and STATUS = ‘Y’.

SELECT BOOKID, TITLE, PRICE

FROM BOOK

WHERE GENRE ='Programming' AND PRICE <=25 AND publisher = 'Head First' and BOOKID IN (SELECT BOOKID

FROM ORDERLINE

WHERE ORDERID IN (SELECT ORDERID

FROM ORDERS

WHERE STATUS= 'Y'));

BOOKID TITLE PRICE

-------------------- ---------------------------------------------------------------------------------------------------- -----

1 Java Developer 25

3 Java Developer 25

2 rows selected.

List of payment which was made by check and in Cork city

SELECT PaymentID, Amount

FROM PAYMENT

WHERE ORDERID IN (SELECT ORDERID

FROM ORDERS WHERE PAYMENT\_METH='Check'

AND CUSTOMERID IN (SELECT CUSTOMERID

FROM CUSTOMER

WHERE CITY= 'Cork'));

PAYMENTID AMOUNT

------------------------- -----

7 10

1 row selected**.**

Viewing records from a book without knowing exact details

SELECT \* FROM BOOK WHERE GENRE LIKE 'P%g';

BOOKID PRICE TITLE AUTHOR GENRE PUBLISHER REL\_DATE PAGE\_

-------------------- ----- ---------------------------------------------------------------------------------------------------- -------------------------------------------------- ------------------------- ---------------------------------------------------------------------------------------------------- --------- -----

1 25 Java Developer Peter Trost Programming Head First 02-MAY-05 200

2 25 Java Developer Peter Trost Programming Head First 02-MAY-05 200

3 25 Java Developer Peter Trost Programming Head First 02-MAY-05 200

4 25 Java Developer Peter Trost Programming Head First 02-MAY-05 200

5 25 Java Developer Peter Trost Programming Head First 02-MAY-05 200

6 15 Android Trust Programming Head First 02-MAY-05 200

7 35 HTML Kust Programming Head First 02-MAY-05 200

8 20 SQL PLUS Zmust Programming Head First 02-MAY-05 200

10 12 MySQL Prust Programming Head First 02-MAY-05 200

9 rows selected.

Select customers who create an order

SELECT CNAME, CITY, ORDER\_DATE

FROM CUSTOMER

LEFT JOIN ORDERS

ON CUSTOMER.CUSTOMERID = ORDERS.CUSTOMERID

INTERSECT

SELECT CNAME, CITY, ORDER\_DATE

FROM CUSTOMER

RIGHT JOIN ORDERS

ON CUSTOMER.CUSTOMERID = ORDERS.CUSTOMERID

CNAME CITY ORDER\_DAT

------------------------- ------------------------- ---------

Alex Dublin 08-APR-15

Alex Dublin 08-DEC-15

Bob Cork 08-AUG-15

John Longford 08-MAY-15

4 rows selected.

# 

# Security

**Manager (ADUNPHY)**

GRANT ALL ON BOOK TO ADUNPHY;

GRANT ALL ON STAFF TO ADUNPHY;

GRANT ALL ON CUSTOMER TO ADUNPHY;

GRANT ALL ON PAYMENT TO ADUNPHY;

GRANT ALL ON ORDERS TO ADUNPHY;

GRANT ALL ON DELIVERY TO ADUNPHY;

GRANT ALL ON ORDERLINE TO ADUNPHY;

GRANT ALL ON COURIER TO ADUNPHY;

GRANT ALL ON TRANSPORT TO ADUNPHY;

**ADMINISTRATOR (W20061961)**

GRANT ALL ON BOOK TO W20061961;

GRANT ALL ON STAFF TO W20061961;

GRANT ALL ON CUSTOMER TO W20061961;

GRANT ALL ON PAYMENT TO W20061961;

GRANT ALL ON ORDERS TO W20061961;

GRANT ALL ON DELIVERY TO W20061961;

GRANT ALL ON ORDERLINE TO W20061961;

GRANT ALL ON COURIER TO W20061961;

GRANT ALL ON TRANSPORT TO W20061961;

**STAFF (W20062427)**

GRANT SELECT, INSERT ON STAFF TO W20062742 WITH GRANT OPTION;

GRANT ALL ON ORDERLINE TO W20062742;

GRANT SELECT, UPDATE ON COLLECTIONPOINT TO W20062742;

**ACCOUNTANT (MLyng)**

GRANT SELECT, INSERT ON PAYMENT TO MLyng;

**COURIER (W20022369)**

GRANT SELECT ON ORDERS TO W20022398 WITH GRANT OPTION;

GRANT SELECT ON DELIVERY TO W20022398;

# CONCLUSIONS

In this paper, I have drafted a database "Web Shop." An analysis of the task with its consistent solution. To develop a database was selected among the SQL 10g. It was reviewed and digested proposed theoretical material, which allowed more efficiently cope with the goals and objectives:

* Studied the subject area
* Built infological model
* Tables were filled with the necessary data
* Created auxiliary elements to work with the database (queries);
* Adding security level access for different types of employees.