**Assignment 3**

**Delitaste Pizza Project**

**Physical Database Design**

**Table of Contents**

Preface: ................................................................................................................................. 2

Data Requirements ................................................................................................................ 2 Transaction Requirements ..................................................................................................... 3 Business Rules ....................................................................................................................... 4 EER Model with Data Dictionary........................................................................................... 6

EER Model ........................................................................................................................ 6 - 7 Data Dictionary ...................................................................................................................... 8 Entities ......................................................................................................................... 8 Relationships ................................................................................................................ 9 Attributes .............................................................................................................. 9 - 12

EER model to the Relational model .............................................................................. 12 - 14

Normalize the schema to Boyce-Codd Normal Form ................................................... 15 - 18

**Assignment 3**: **Delitaste Pizza Project**

**School of Information and Physical Sciences - COMP6140 University of Newcastle, Australia**

**Data Requirements:**

***Order***

Order is the process of the necessary information to take a customer’s order. Orders can be divided in two types: phone and walk-in. The data stored regarding the order includes the date & time, employee id number, order type (phone & walk-in), customer phone number, its details, the items ordered (items name, item code, quantity, and price), the total amount due, order status, description and payment method (Credit Card/Debit Card/Cash). When the payment is by card, the payment approval number is also recorded.

With phone orders requesting pick up, the pick-up time will be recorded. For walk-in orders only pickup is available, the pickup time is then stored.

For Delivery orders the data stored are delivery time, address (unit/street number, name and suburb) and the driver who delivered the order.

***Customer***

Customers are people who have used the store system to order or who are going to order from a Delitaste Pizza’s shop. The details of the data stored regarding a customer are customerID, phone number, first name, last name, address (unit/street number, name, suburb and postcode) and status (orders history).

***Staff***

Staff are the employees of the Delitaste Pizza shop. Employees can be divided in two types: those who work in the shop (Instore) and those who carry out deliveries (Driver). The data stored regarding each staff member is their employee number (Id), name (first and last), postal address (number/street number, street name, suburb and postcode), contact number, tax file number, bank details (bank code, bank name, account number), status, and description.

For instore staff, data stored includes hourly rate. Extra data is required for Drivers, such as driver’s license number.

***Shift***

Shift is when a staff member works a period of hours for the company. It can be divided in two types: Drivers shift and Instore shift. The data recorded on each shift is recordId, start date, start time, end date, end time and the orders.

For a Drivers shift, data regarding delivered orders is also recorded.

***Payment***

Payment is the process to perform both Instore and Drivers employee’s salary payments. It is divided in two types, InstorePay and DriversPay. The data stored on each payment includes payment record id, gross payment, tax withheld, total amount paid, payment date, bank details of the employee, payment period starts date and payment period end date.

For InstorePay the paid hourly rate is recorded and for the DriversPay the paid number of deliveries is stored.

***MenuItem***

Menu Items are the combined description of items sold in the shop. Each item in the menu has a unique code. The data stored includes item number, name, size(small/medium/large) and a current selling price.

***Ingredients***

The items in the menu are made up of a number of ingredients. The data stored for each ingredient includes a code (unique), name, type, description, stock level at current stocktake period, date last stocktake was taken, suggested current stock level, reorder level.

***IngredientOrder***

Ingredients Order is the process where the store manager uses the actual levels of ingredients in store, together with suggested current stock levels and reorder levels to order ingredients for the following week. The data stored about ingredient orders are order number, date of the order, date received order, total amount, order status, description, Ingredient code, quantity and price of all ingredients.

**Transaction Requirements:**

***Data Manipulation Operations***

\* Insert/update/delete the details of Order

\* Insert/update/delete the details of PhoneOrders

\* Insert/update/delete the details of Walk-inOrders

\* Insert/update/delete the details of DeliveryOrders

\* Insert/update/delete the details of Pick-upOrders

\* Insert/update/delete the details of Customers

\* Insert/update/delete the details of InstoreStaff

\* Insert/update/delete the details of DriverStaff

\* Insert/update/delete the details of Staff

\* Insert/update/delete the details of Payment

\* Insert/update/delete the details of DriverPayment

\* Insert/update/delete the details of InstorePayment

\* Insert/update/delete the details of Shift

\* Insert/update/delete the details of MenuItem

\* Insert/update/delete the details of Ingredients

\* Insert/update/delete the details of IngredientOrder

***Data Queries***

\*Search a delivery based on a customer order number

\*List delivery orders, on a particular date

\*List customers order based on their phone number and/or name

\*List orders based on staff number and date

\*List phone orders paid by card

\*List the customer’s name, address and total spent in descending order

\*List the total orders of a specific customer from last year

\*List the name and address of all customers living in the local shop area

\*List orders on hold

\*List all details from customers who purchased yesterday.

\*List the menu item name that sold the most in the last 3 month

\*List menu items by total sales in descending order

\*List menu items sold in the last Christmas Evening

\*List all Ingredients

\*List the most ordered Ingredients in the last month

\*List the ingredients delivered yesterday

\*List the count of pick-up orders last year

\*Search menu items by code number

\*List ordered menu items, on a particular date

\*List ingredients and quantity used for menu items

\*Report of ingredient levels for current period and suggested stock levels

\*List Ingredients ordered based on order number and/or date

\*List staff members by type, shift and date

\*List the driver shift delivery orders from last month

\*List the drivers and Instore staffs who worked the last public holiday

\*Search a payment based on an employee number, name, on a particular pay date.

\*Search Staff members based on employee number and name

\*List the Staff names and numbers of all staff working more than a year in the company

**Business Rules:**

***Orders***

* There can only be two types of orders:
* Pick up
* Delivery
* When a customer orders, the customer’s phone number must be entered into the system along with the id of the staff taking the order.
* After a phone order has been taken, a verification process must occur whereby the staff dials the number given and confirms the order with the customer.
* If order is not confirmed, the customer is recorded as a hoax and the order is kept on hold (if and until the customer calls back in which case the verification process takes place before the hoax is removed and the order goes through).
* For each phone order, the time the call was answered as well as the time the call was terminated must be recorded.
* For delivery order the delivery time and address and the driver who delivered the order must be recorded.
* For each walk-in order, the time the customer walks in is recorded. Only pickup is available, and the pickup time must be recorded.
* The orders payment method can only have one of the following values:
* Credit Card
* Debit Card
* Cash
* When an order is paid by card, the approval number must be stored in the order’s PaymentApprovalNo.
* The orders a driver delivers during a shift must be kept on the record.

***Customer***

* A new customer must be marked as un-verified until the verification process is successfully completed.

***Employee***

* An employee can only be either an in-store worker or a delivery driver.
* Employees must record each shift they work in the database.
* In-store staffs are paid hourly
* Delivery drivers are paid by the number of deliveries.
* Employee payment record must be maintained in the database.

***Menu Items, Ingredients, and Ingredient Order***

* MenuItems must have its unique identification number.
* The amount of each ingredient remaining must be updated every time some is used.
* A stocktake must be taken each week, where the actual levels of ingredients in store are collected.
* The results of the weekly stocktake must be input into the database.
* When an Ingredient stock level decreases below its reorder level an order for the ingredient must be placed.

**EER Model with data dictionary**

****

**EER model (Entities and Attributes)**

|  |
| --- |
| Order |
| OrderNo {PK}  Date&Time  StaffId  Type  Phone  Walk-in  Pick-up  Time  Delivery  DeliveryTime  DeliveryAddress  Unit/StreetNo  Street  Suburb  MenuItemCode  Name  Price  Quantity  TotalAmountDue PayMethod  CreditCard  PaymentApprovalNo  DebitCard  Cash  Status  Description |

|  |
| --- |
| Customer |
| CustomerID {PK}  PhoneNumber  Name  First  Last  Address  Unit/StreetNo  Street  Suburb  PostCode  Status |

|  |
| --- |
| Staff |
| Id {PK}  Name  First  Last  PostalAddress  Unit/No  Street  Suburb  PostCode  ContactNumber  TaxFileNo  BankDetails  BankCode  Name  AccNo  Status  Description  Type  InStore  HourRate  Driver  Del.Rate  DriverLicense |

|  |
| --- |
| IngredientOrder |
| OrderNo {PK}  Date  ReceivedDate  Status  Description  Quantity  TotalAmount |

|  |
| --- |
| Payment |
| RecordNo {PK}  DriverPay  InstorePay  StaffId  GrossPayment  TaxWithheld  TotalAmountPaid  Date  PeriodStartDate  PeriodEndDate |

|  |
| --- |
| MenuItem |
| ItemCode {PK}  Name  Size  Small  Medium  Large  Price |

|  |
| --- |
| Ingredients |
| Code {PK}  Name  Type  Description  StockLevel DateLastStocktake SuggestedStockLevel ReorderLevel |

|  |
| --- |
| Shift |
| RecordId {PK}  StartDate  StartTime  EndDate  EndTime  InstoreShift  DriverShift  DeliveryOrdersNo |

**Data Dictionary:**

***Entities***

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity Name** | **Description** | **Aliases** | **Occurrence** |
| Order | General term describing orders customers have made | Pizza  order | When an order is made by a customer |
| DeliveryOrder | General term describing customer delivery orders |  | When an order is going to be deliver |
| Pick-upOder | General term describing customer pick-up order |  | Walk in customers or phone order request |
| Customer | General term describing the user of our system who order or is going to order | Client | When a person calls or walk-in to order |
| Staff | General term describing an employee that work | Employee | Those who work for the shop (company) |
| InstoreStaff | General term describing staff working in store |  | Staff who perform works on in store |
| DriverStaff | General term describing staff working as delivery drivers |  | Delivery orders are assigned to a driver |
| Payment | General term describing staff payment | Salary | After a shift finishes a payment is generated |
| DriverPay | General term describing member of staff driver payment |  | After a driver finishes his/her shift |
| InstorePay | General term describing Instore staff payment |  | After a Instore staff finishes his/her shift |
| Shift | General term describing the period of work in the shop |  | During the shop trading hours |
| MenuItem | General term describing items for sale in the shop | Product  list | When placed an order one or more item of the menu is selected |
| Ingredients | General term describing ingredients of each MenuItem | Pizza  components | An item in the menu is made up of a number of ingredients |
| IngredientOrder | General term describing the supply orders of ingredients | Stock order | When the store manager orders ingredients for the following week |

***Relationship types***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Multiplicity** | **Relationship** | **Multiplicity** | **Entity Name** |
| Order | (Man, Or) | Generalization | (Man, Or) | Walk-inOrder |
|  | (Man, Or) | Generalization | (Man, Or) | PhoneOrder |
|  | 0..\* | Has | 1..\* | MenuItem |
| PhoneOrder | (Man, Or) | Generalization | (Man, Or) | Pick-upOrder |
|  | (Man, Or) | Generalization | (Man, Or) | DeliveryOrder |
| DeliveryOrder | 1..\* | Informs | 1..1 | DriverShift |
| Customer | 1..1 | Has | 1..\* | Order |
|  | (Man, Or) | Generalization | (Man, Or) | InStoreStaff |
| Staff | (Man, Or) | Generalization | (Man, Or) | DriverStaff |
| InstoreStaff | 1..1 | Takes | 0..1 | Order |
|  | 1..\* | Works | 0..\* | InStoreShift |
| DriverStaff | 1..1 | Delivers | 0..\* | DriverShift |
| MenuItem | 1..\* | Made of | 1..\* | Ingredients |
| Ingredients | 0..\* | Controls Stock | 0..\* | IngredientOrder |
| Shift | (Man, Or) | Generalization | (Man, Or) | DriverShif |
|  | (Man, Or) | Generalization | (Man, Or) | InstoreShift |
| DriverShift | 0..\* | Informs | 1..1 | DriverPay |
| InstoreShift | 0..\* | Informs | 1..1 | InstorePay |
| Payment | (Man, Or) | Generalization | (Man, Or) | DriverPay |
|  | (Man, Or) | Generalization | (Man, Or) | InstorePay |

***Attributes***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Attributes** | **Description** | **Data Type & Length** | **Nulls** | **Multi-valued** | **Deri-ved** | **Defult\*** |
| Order | OrderNo | Unique order identifier | varchar(10) | N | N | N | None |
|  | Order Date&Time | The date & time the order is made | datetime | N | N | N | None |
|  | OrderType | Walk-in or Phone order | varchar(10) | N | N | N | None |
|  |  |  |  |  |  |  |  |
|  | PhoneNo | Customer phone number with details | varchar(10) | N | N | Y | None |
|  | OrderPick-up | Orders that will be picked-up | 1 varchar (Y/N) | Y | N | N | None |
|  | Time | Time order is picked-up | time | Y | N | Y | None |
|  | OrderDelivery | Requested delivery orders | 1 varchar (Y/N) | Y | N | N | None |
|  | Time | The time an order was delivered | time | Y | N | N | None |
|  | Address    Unit/No  StName  Suburb | Delivery address  Unit & Street number  Street name  Suburb name | varchar(3)  varchar(5)  varchar(20)  varchar(20) | Y  Y  Y  Y | N  N  N  N | N  N  N  N | None  None  None  None |
|  |  |  |  |  |  |  |  |
|  | Quantity | Ordered item quantity | varchar(5) | N | N | Y | None |
|  | TotalAmount | Total sum of item prices | float | N | N | N | None |
|  | PayMethod  ApprovalNo | Method of payment  ApprovalNo | varchar(20)  char(10) | N  Y | N  N | N  N | None  None |
|  | Status | Customer Order history | char(15) | Y | N | N | None |
|  | Description | Additional order description | char(20) | Y | N | N | None |
| Customer | CustomerID  PhoneNumber | Unique Customer Id  Customer phone number | char(4)  varchar(10) | N  N | N  N | N  N | None  None |
|  | Name  First  Last | First name of customer  Last name of customer | varchar(15)  varchar(15) | N  N | N  N | N  N | None  None |
|  | Address  Unit/No    StName  Suburb  PostCode | Customer address  Unit and street number  Street name  Suburb name  Postal Code | varchar(3)  varchar(5)  varchar(20)  varchar(15)  varchar(4) | Y  N  N  Y  N | N  N  N  N  N | N  N  N  N  N | None  None  None  None  None |
|  | Status | Customer orders history occurrences | char(30) | Y | N | N | None |
| Staff | StaffId  Name  First  Last  PostalAddress  Unit/No  Street  Suburb  PostCode  ContactNumber  TaxFileNo  BankDetails  BankCode  bName  AccNo  Status  Description  HourRate  DeliverRate  DriverLicense | Unique Staff identifier  Staff first name  Staff last name  Staff postal address  Unit number &  Street number  Street name  Suburb name  Postal code  Staff phone contact number  Staff tax file number  Staff bank details  Staff bank code  Staff bank name  Account number  Staff status  Staff description  Staff hourly rate  Delivery rate  Staff driver’s license | varchar(4)  varchar(15)  varchar(15)  varchar(3)  varchar(5)  varchar(20)  varchar(15)  varchar(4)  varchar(10)  varchar(12)  varchar(6)  varchar(15)  varchar(10)  varchar(15)  varchar(20)  varchar(5)  varchar(10)  varchar(8) | N  N  N  Y  N  N  Y  N  N  Y  N  N  N  Y  Y  N  N  Y | N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N | N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N  N | None  None  None  None  None  None  None  None  None  None  None  None  None  None  None  None  None  None |
| Payment | RecordNo  GrossPayment  TaxWithheld  TotalAmountPaid  Date  PeriodStarDate  PeriodEndDate | Payment record number  Payment gross total  Payment tax total withheld  Payment total amount paid  Payment Date  Payment date of start period  Payment date of end period | cha(5)  varcha(10)  varcha(10)  varcha(10)  date  date  date | N  N  N  N  N  N  N | N  N  N  N  N  N  N | N  N  Y  Y  N  N  N | None  None  None  None  None  None  None |
| Shift | RecordId  StartDate  StartTime  EndDate  EndTime | Shift record Id  Shift starting date  Shift starting time  Shift ending date  Shift ending time | char(5)  date  time  date  time | N  N  N  N  N | N  N  N  N  N | N  N  N  Y  Y | None  None  None  None  None |
| MenuItem | ItemCode  Name  Size    Small  Medium  Large  Price | Menu item unique code identifier  Menu item name  Menu item size click box to choose  pizza small  pizza medium M  pizza large L  Menu item price | varchar(6)  varchar(15)  varchar(1)  varchar(1)  varchar(1)  varchar(6) | N  N  Y  Y  Y  N | N  N  N  N  N  N | N  N  Y  Y  Y  Y | None  None  None  Med  None  None |
| Ingredients | Code  Name  Type  Description  StockLevel  DateLastStockTake SuggestedStockLevel ReorderLevel | Ingredient unique code identifier  Ingredient name  Ingredient type  Ingredient description  Ingredient level of stock  Ingredient date of last stock took  Ingredient suggested stock level  Ingredient stock reorder level | varchar(10)  varchar(40)  varchar(40)  varchar(40)  varchar(10)  date  varchar(10)  varchar(10) | N  N  Y  N  N  N  Y  Y | N  N  N  N  N  N  N  N | N  N  N  N  Y  N  N  N | None  None  None  None  None  None  None  Y |
| IngredientOrder | OrderNo  Date  ReceivedDate  Status  Description  Quantity  TotalAmount | Ingredients order number  Ingredients order date  Ingredient order received date  Ingredient order status  Ingredient order description  Ingredient order quantity  Ingredient order total amount value | varchar(10)  date  date  varchar(15)  varchar(25)  varchar(5)  float | N  N  Y  Y  Y  N  N | N  N  N  N  N  N  N | N  N  N  N  N  Y  N | None  None  None  None  None  None  None |

* 1. **Map the EER model to the relational model. Document the relational schema in DBDL**

Rule 1 is: for super/subclass, Mandatory &Disjoint: create multi-relations, with 1 relation for each combined superclass/subclass;

Rule 2 is: \*to\* relation type (for Order has MenuItem and MenuItem made of Ingredientes): create “Relationship” relation and two foreign keys;

Rule 3 is: 1 to \* relation type (for InstoreStaff takes Order): create a foreign key at Order side.

**Orders(**OrderNo, OrderDate&Time, OrderType, CustomerID, StaffId, TotalAmountDue, PaymentMethod, PaymentApprovalNo, Status**)**

**Primary Key** OrderNo

**Foreign Key** CustomerID **references** Customer(CustomerID)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** StaffId **references** InStoreStaff(StaffId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**PhoneOrder(**OrderNo, OrderDate&Time,**)**

**Primary Key** OrderNo

**Foreign Key** OrderNo **references** Orders(OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**WalkinOrder(**OrderNo, WalkInTime**)**

**Primary Key** OrderNo

**Foreign Key** OrderNo **references** Orders(OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**PickupOrder(**OrderNo, OrderPickupTime**)**

**Primary Key** OrderNo

**Foreign Key** OrderNo **references** Orders(OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**DeliveryOrder(**OrderNo, DeliveryTime, Unit/No, StreetName, Suburb, ShiftRecordId**)**

**Primary Key** OrderNo

**Foreign Key** OrderNo **references** PhoneOrder(OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** ShiftRecordId **references** DriverShift (RecordId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Customer(**CustomerID**,** PhoneNo, fName, lName, Unit/No, StreetName, Suburb, Postcode, Status**)**

**Primary Key** CustomerID

**Alternative Key** lName

**InstoreStaff(**StaffId, fName, lName, Unit/No, StreetName, Suburbe, Postcode, ContactNo, TaxFileNo, BankCode, bName, accNo, Status, HourRate**)**

**Primary Key** StaffId

**Alternative Key** lName

**DriverStaff(**StaffId, fName, lName, Unit/No, StreetName, Suburbe, Postcode, ContactNo, taxFileNo, BankCode, bName, accNo, Status, DeliveryRate, DriverLicense**)**

**Primary Key** StaffId

**Alternative Key** lName

**DriverShift(**RecordId, StartDate, StartTime, EndDate, EndTime, StaffId, DriverPayRecordId**)**

**Primary Key** RecordId

**Foreign Key** StaffId **references** DriverStaff (StaffId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** DriverPayRecordId **references** DriverPay(RecordId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**InstoreShift(**RecordId, StartDate, StartTime, EndDate, EndTime, StaffId, InstorePayRecordId**)**

**Primary Key** RecordId

**Foreign Key** StaffId **references** InstoreStaff (StaffId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** InstorePayRecordId **references** InstorePay (RecordId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**DriverPayment(**RecordId, GrossPayment, TaxWithheld, TotalAmountPaid, Date, PeriodStartDate, PeriodEndDate**)**

**Primary Key** RecordId

**InstorePayment(**RecordId, GrossPayment, TaxWithheld, TotalAmountPaid, Date, PeriodStartDate, PeriodEndDate**)**

**Primary Key** RecordId

**MenuItem(**ItemCode, Name, Small, Medium, Large, Price**)**

**Primary Key** ItemCode

**Alternative Key** Name

**Ingredients(**Code, Name, Type, Description, StockLevel, DateLastStockTake, SuggestedStockLevel, ReorderLevel**)**

**Primary Key** Code

**Alternative Key** Name

**IngredientOrder(**OrderNo, Date, ReceivedDate, Status, Description, quantity, TotalAmount**)**

**Primary Key** OrderNo

**QuantityOrderMenuItem(**OrderNo, quantity, ItemCode**)**

**Primary Key** OrderNo, ItemCode

**Foreign Key** OrderNo **references** Order (OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** ItemCode **references** MenuItem (ItemCode)

**ON UPDATE CASCADE ON DELETE CASCADE**

**MenuItemMadeofIngredientes(**ItemCode, quantity, Code**)**

**Primary Key** ItemCode, Code

**Foreign Key** ItemCode **references** MenuItem (ItemCode)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** Code **references** Ingredients (Code)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**IngredientsQuantityIngredientOrder(**Code, quantity, OrderNo**)**

**Primary Key** Code, OrderNo

**Foreign Key** Code **references** Ingredients(Code)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** OrderNo **references** IngredientOrder (OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

* 1. **Normalize the schema to Boyce-Codd Normal Form (if any relation is not already in BCNF). The final normalised schema must be documented in DBDL.**

**Relational Schema in BCNF Form**

**Orders(**OrderNo, OrderDate&Time, OrderType, CustomerID, StaffId, TotalAmountDue, PaymentMethod, PaymentApprovalNo, Status**)**

**Primary Key** OrderNo

**Foreign Key** CustomerID **references** Customer(CustomerID)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** StaffId **references** InStoreStaff(StaffId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

FD1: **OrderNo ->** OrderDate, OrderTime, OrderType, StaffId, CustomerID, PaymentApprovalNo, Status  
FD2: **PaymentApprovalNo ->** PaymentMethod, TotalAmountDue

There exists transitive dependency, so the relation is in 2nd but not 3rd NF.

Normalization conversion in 3rd NF:

**OrderOnly** (OrderNo, OrderDate&Time, OrderType, PaymentApprovalNo, Status)

**Primary Key** OrderNo

**Foreign Key** PaymentApprovalNo **references**OrderPayRecord(PaymentApprovalNo) **ON UPDATE CASCADE, ON DELETE CASCADE**

**OrderPayRecord** (PaymentApprovalNo, PaymentMethod, TotalAmountDue)

**Primary Key** PaymentApprovalNo

--This table is in BCNF. This is in 3rd normal form. Because there is no functional dependency.

**PhoneOrder(**OrderNo, OrderDate&Time,**)**

**Primary Key** OrderNo

**Foreign Key** OrderNo **references** Orders(OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**WalkinOrder(**OrderNo, WalkInTime**)**

**Primary Key** OrderNo

**Foreign Key** OrderNo **references** Orders(OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**PickupOrder(**OrderNo, OrderPickupTime**)**

**Primary Key** OrderNo

**Foreign Key** OrderNo **references** Orders(OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**DeliveryOrder(**OrderNo, DeliveryTime, Unit/No, StreetName, Suburb, ShiftRecordId**)**

**Primary Key** OrderNo

**Foreign Key** OrderNo **references** PhoneOrder(OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** ShiftRecordId **references** DriverShift (RecordId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Customer(**CustomerID**,** PhoneNo, fName, lName, Unit/No, StreetName, Suburb, Postcode, Status**)**

**Primary Key** CustomerID

**Alternative Key** lName

**InstoreStaff(**StaffId, fName, lName, Unit/No, StreetName, Suburbe, Postcode, ContactNo, TaxFileNo, BankCode, bName, accNo, Status, HourRate**)**

**Primary Key** StaffId

**Alternative Key** lName

**DriverStaff(**StaffId, fName, lName, Unit/No, StreetName, Suburbe, Postcode, ContactNo, taxFileNo, BankCode, bName, accNo, Status, DeliveryRate, DriverLicense**)**

**Primary Key** StaffId

**Alternative Key** lName

**DriverShift(**RecordId, StartDate, StartTime, EndDate, EndTime, StaffId, DriverPayRecordId**)**

**Primary Key** RecordId

**Foreign Key** StaffId **references** DriverStaff (StaffId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** DriverPayRecordId **references** DriverPay(RecordId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**InstoreShift(**RecordId, StartDate, StartTime, EndDate, EndTime, StaffId, InstorePayRecordId**)**

**Primary Key** RecordId

**Foreign Key** StaffId **references** InstoreStaff (StaffId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** InstorePayRecordId **references** InstorePay (RecordId)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**DriverPayment(**RecordId, GrossPayment, TaxWithheld, TotalAmountPaid, Date, PeriodStartDate, PeriodEndDate**)**

**Primary Key** RecordId

FD1: **RecordId->** TotalAmountPaid, Date, PeriodStartDate, PeriodEndDate

FD2: **TotalAmountPaid - >** GrossPayment, TaxWithheld

There exists transitive dependency, so the relation is in 2nd but not 3rd NF.

Normalization conversion in 3rd NF:  
**DriverPayOnly(**RecordId,TotalAmountPaid, Date, PeriodStartDate, PeriodEndDate**)**

**Primary Key RecordId**

**Foreign Key** TotalAmountPaid references DriverPayRecord(TotalAmountPaid) **ON UPDATE CASCADE, ON DELETE CASCADE**

**DriverPayRecord(**TotalAmountPaid, GrossPayment, TaxWithheld**)**

**Primary Key** TotalAmountPaid

--This table is in BCNF. This is in 3rd normal form. Because there is no functional dependency.

**InstorePayment(**RecordId, GrossPayment, TaxWithheld, TotalAmountPaid, Date, PeriodStartDate, PeriodEndDate**)**

**Primary Key** RecordId

FD1: **RecordId->** TotalAmountPaid, Date, PeriodStartDate, PeriodEndDate

FD2: **TotalAmountPaid - >** GrossPayment, TaxWithheld

There exists transitive dependency, so the relation is in 2nd but not 3rd NF.

Normalization conversion in 3rd NF:  
**InstorePayOnly(**RecordId,TotalAmountPaid, Date, PeriodStartDate, PeriodEndDate**)**

**Primary Key RecordId**

**Foreign Key** TotalAmountPaid references DriverPayRecord(TotalAmountPaid) **ON UPDATE CASCADE, ON DELETE CASCADE**

**InstorePayRecord(**TotalAmountPaid, GrossPayment, TaxWithheld**)**

**Primary Key** TotalAmountPaid

--This table is in BCNF. This is in 3rd normal form. Because there is no functional dependency.

**MenuItem(**ItemCode, Name, Small, Medium, Large, Price**)**

**Primary Key** ItemCode

**Alternative Key** Name

FD1: ItemCode-> Name, Price

FD2: Price -> Small

FD3: Price-> Medium

FD4: Price ->Large

**MenuItem**(ItemCode, Name, Size, Price, CurrentSellingPrice)

**Primary Key** ItemCode

**Foreign Key**CurrentSellingPrice**references**MenuSellingPrice(CurrentSellingPrice)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**MenuSellingPrice(**CurrentSellingPrice, Small, Medium, Large**)**

**Primary Key** CurrentSellingPrice

--This table is in BCNF

**Ingredients(**Code, Name, Type, Description, StockLevel, DateLastStockTake, SuggestedStockLevel, ReorderLevel**)**

**Primary Key** Code

**Alternative Key** Name

FD1:Code-> Name, Type, Description, StockLevel, DateLastStockTake

FD2:StockLevel-> SuggestedStockLevel, ReorderLevel

**IngredientsOnly**(Code, Name, Type, Description, StockLevel, DateLastStockTake)

**Primary Key** Code

**Foreign Key** StockLevel **references** IngredientsStock(Stocklevel)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**IngredientsStock(**StockLevel, SuggestedStockLevel, ReorderLevel**)**

**Primary Key** StockLevel

--This table is in BCNF

**IngredientOrder(**OrderNo, Date, ReceivedDate, Status, Description, quantity, TotalAmount**)**

**Primary Key** OrderNo

**QuantityOrderMenuItem(**OrderNo, quantity, ItemCode**)** --This table is in BCNF

**Primary Key** OrderNo, ItemCode

**Foreign Key** OrderNo **references** Order (OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** ItemCode **references** MenuItem (ItemCode)

**ON UPDATE CASCADE ON DELETE CASCADE**

**MenuItemMadeofIngredientes(**ItemCode, quantity, Code**)** --This table is in BCNF

**Primary Key** ItemCode, Code

**Foreign Key** ItemCode **references** MenuItem (ItemCode)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** Code **references** Ingredients (Code)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**IngredientsQuantityIngredientOrder(**Code, quantity, OrderNo**)** --This table is in BCNF

**Primary Key** Code, OrderNo

**Foreign Key** Code **references** Ingredients(Code)

**ON UPDATE CASCADE, ON DELETE CASCADE**

**Foreign Key** OrderNo **references** IngredientOrder (OrderNo)

**ON UPDATE CASCADE, ON DELETE CASCADE**