

Portelli Wine Merchants

Accounting and Inventory Database Design

DG504 – Database Design and Queries

Table of Contents

ANALYSIS.....	3 - 4
Data Requirements	3
Data Model	4
DESIGN.....	5 - 8
Candidate Relationships	5
Normalised Relations	5
Data Dictionary	6 - 8
CREATE TABLE	9 - 10
MODEL DATA.....	11 - 13
SELECT STATEMENTS.....	14 - 15
Testing Queries	14
Insert Query	15
Update Query	15
Delete Query	15

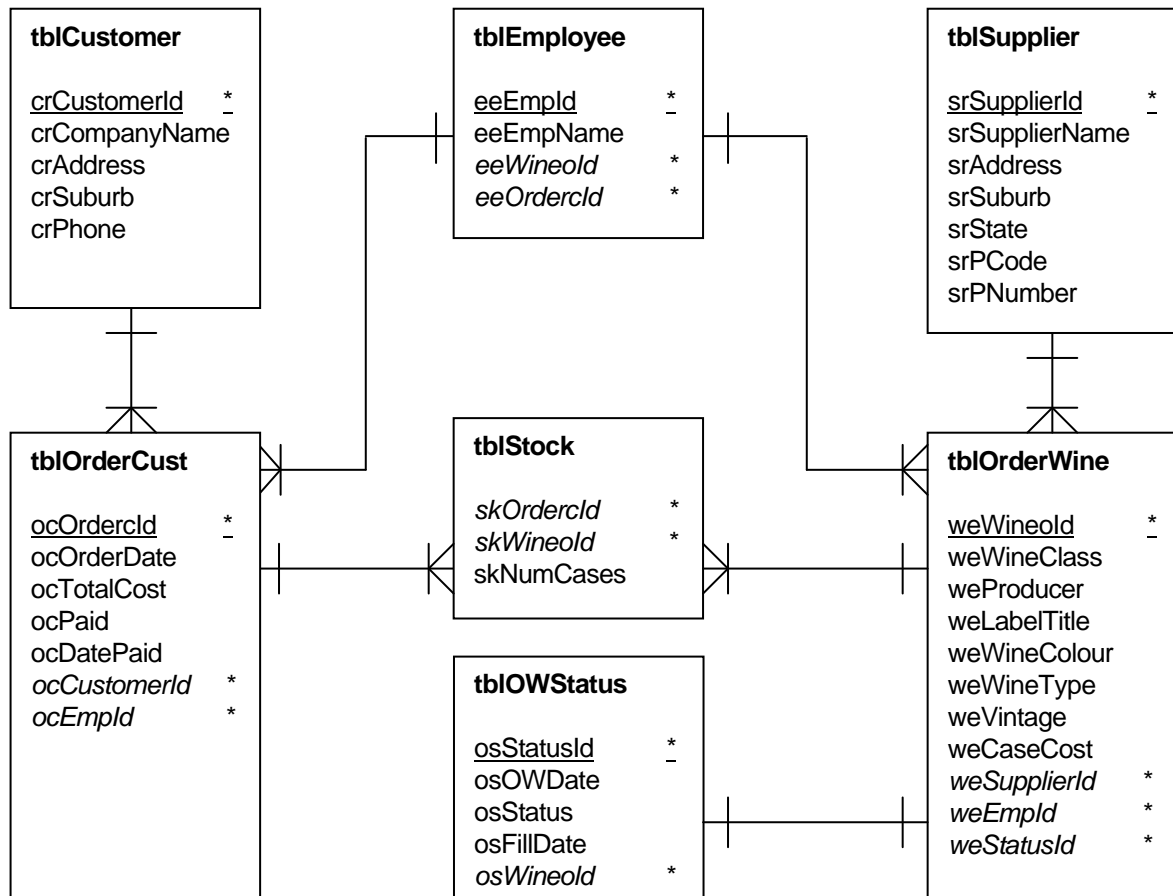
ANALYSIS // Data Requirements

Analyse the data required for an application to establish the data requirements.

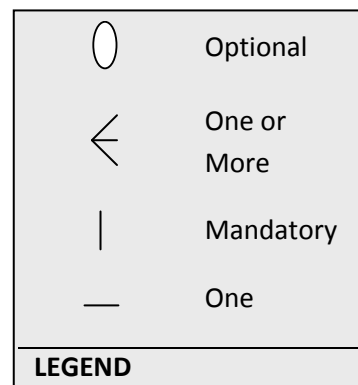
- Defined Entities: Customer, Customer Order, Stock, Wine Order, Status, Employee, Suppliers
 - The hotel/pub (customer) may submit one or more orders at a time
 - Each customer wine order must have only one customer
 - Customer wine orders are fulfilled based on Stock
 - Stock unavailable must be ordered through Supplier
 - Wines must have one or more stock available
 - Wines must have a supplier
 - Employee IDs must be on orders responsible
 - Wine orders must have a status
 - A Status can be associated to one or more wine order
 - Suppliers can provide one or more wine types

- The Wine is sold to local retail outlets in Melbourne who are regular customers.
- Orders are directly entered into the computer with each new customer order.
- Portelli staff must be assigned to customer and wine orders for difficulties contact and tracking.
- Customer Order payments can be doing within a time frame, checked off when paid.
- Wines are all held at St. Albans therefore inadequate stocks need only be ordered for a single location from suppliers.
- Each supplier may supply several wine types, but each wine is supplied by only one supplier is the current company policy.
- Ordered Wine from supplies are to be logged with each request and whether it has been fulfilled.

ANALYSIS // Data Model (Entity Relationship Diagram)



Structure the data to create a data model graphically represented by an Entity-Relationship(ER) diagram and augmented with Entity-Attributed (EA) worksheets.



DESIGN // Candidate Relationships

Derive the candidate relations. Transform the entity model to a relational model.

CR

Customer (CustomerId, CompanyName, ContactName, Address, Suburb, Phone)

Employee (EmpId, EmpName)

OrderCust (OrderId, CustomerId, EmpId, OrderDate, TotalCost, Paid, DatePaid,)

OrderWine (WineId, SupplierId, EmpId, WineClass, Producer, LabelTitle, WineColour, WineType, Vintage, CaseCost, , StatusId)

OWStatus (StatusId, status, OWDate, FillDate)

Stock (OrderID, WineId, NumCases)

Supplier (SupplierId, SupplierName, Address, Suburb, State, PCode, PNumber)

DESIGN // Normalised Relations

Normalise the candidate relations to at least third normal form (3NF).

Third Normal Form (3NF)

Customer (CustomerId, CompanyName, ContactName, Address, Suburb, Phone)

Employee (EmpId, EmpName)

OrderCust (OrderId, CustomerId, EmpId, OrderDate, Paid, DatePaid,)

OrderWine (WineId, SupplierId, EmpId, WineClass, Producer, LabelTitle, WineColour, WineType, Vintage, CaseCost, , StatusId)

OWStatus (StatusId, status, OWDate, FillDate)

Stock (OrderID, WineId, NumCases)

Supplier (SupplierId, SupplierName, Address, Suburb, State, PCode, PNumber)

DESIGN // Data Dictionary**tblCustomer**

Field	Contents	Type	Length	Format	Range - Validation	Key / Req
crCustomerId	Unique Customer Identification number	AutoNumber	Long Integer			PK / Y
crCompanyName	Company Name	Text	30	30(X)		Y
crAddress	Company Address	Text	30	30(X)		Y
crSuburb	Company Suburb	Text	30	30(X)		Y
crPhone	Company Phone Number	Text	12	12(X)		Y

tblEmployee

Field	Contents	Type	Length	Format	Range - Validation	Key / Req
eeEmpId	Unique Customer Identification number	AutoNumber	Long Integer			PK / Y
eeEmpName	Employee Name	Text	30	30(X)		Y
eeWineold	Foreign key from Order Wine table	Text	5	XXXXX		FK / Y
eeOrdercld	Foreign key from Order Cust table	Number	Long Integer			FK / Y

tblOrderCust

Field	Contents	Type	Length	Format	Range - Validation	Key / Req
ocOrdercld	Unique Order Identification number	AutoNumber	Long Integer			PK / Y
ocOrderDate	Order Creation Date	Date/Time		Short Date		Y

ocTotalCost	Total Cost of Order	Currency		Currency		Y
ocPaid	Order has been paid or not	Yes/No		Yes/No		Y
ocDatePaid	Order Paid Date	Date/Time		Short Date		N
ocCustomerId	Foreign key from Customer table	Number	Long Integer			FK / Y
ocEmpId	Foreign key from Customer table	Number	Long Integer			FK / Y

tblOWStatus

Field	Contents	Type	Length	Format	Range - Validation	Key / Req
osStatusId	Unique Customer Identification number	AutoNumber	Long Integer			PK / Y
osStatus	Order has been filled	Yes/No		Yes/No		Y
osOWDate	Ordered Wine Date	Date/Time		Short Date		Y
osWineold	Foreign key from Order Cust table	Text	5	XXXXX		FK / Y

tblStock

Field	Contents	Type	Length	Format	Range - Validation	Key / Req
skOrderId	Foreign key from Wine table	Number	Long Integer			FK / Y
skWineold	Foreign key from Wine table	Text	5	XXXXX		FK / Y
skNumCases	Number of Wine Cases	Number	Integer	9999		Y

tblSupplier						
Field	Contents	Type	Length	Format	Range - Validation	Key / Req
srSupplierId	Unique Supplier Identification number	AutoNumber	Long Integer			PK / Y
srSupplierName	Name of Supplier	Text	30	30(X)		Y
srAddress	Supplier Address	Text	30	30(X)		Y
srSuburb	Supplier Suburb	Text	30	30(X)		Y
srState	Supplier State	Text	3	XXX		Y
srPCode	Supplier Postcode	Text	4	XXXX		Y
srPNumber		Text	12	12(X)		Y

tblOrderWine						
Field	Contents	Type	Length	Format	Range - Validation	Key / Req
weWineold	Unique Wine Identification number	Text	5	XXXXX		PK / Y
weWineClass	Class of Wine	Text	30	30(X)		Y
weProducer	Producer Name	Text	30	30(X)		Y
weLabelTitle	Name of Wine	Text	30	30(X)		Y
weWineColour	Colour of Wine	Text	20	20(X)		Y
weWineType	Type of Wine	Text	20	20(X)		Y
weVintage	Production Year	Text	4	XXXX		Y
weCaseCost	Cost per Case	Currency	9999	\$9999		Y
weSupplierId	Foreign key from Supplier table	Number	Long Integer			FK / Y

DESIGN // Create Table

Type appropriate commands to create a minimum of 3 related tables in the data dictionary, at least one of these tables must contain a primary key and one or more foreign keys.

```
CREATE TABLE tblCustomer
(
  crCustomerId AUTOINCREMENT(1) PRIMARY KEY,
  crCompanyName CHAR(30) NOT NULL,
  crAddress CHAR(30) NOT NULL,
  crSuburb CHAR(30) NOT NULL,
  crPhone CHAR(30) NOT NULL );
```

```
CREATE TABLE tblEmployee
(
  eeEmpId AUTOINCREMENT(1) PRIMARY KEY,
  eeEmpName CHAR(30) NOT NULL,
  eeWineold CHAR(5),
  eeOrdercd INT,
  CONSTRAINT fk_EwOrder FOREIGN KEY (eeWineold)
  REFERENCES tblOrderWine(weWineold),
  CONSTRAINT fk_EcOrder FOREIGN KEY (eeOrdercd)
  REFERENCES tblOrderCust(ocOrdercd)
);
```

```
CREATE TABLE tblOrderCust
(
  ocOrderCld AUTOINCREMENT(1) PRIMARY KEY,
  ocOrderDate DATE NOT NULL,
  ocTotalCost CURRENCY NOT NULL,
  ocPaid BIT NOT NULL,
  ocDatePaid DATE,
  ocCustomerId INT,
  ocEmpId INT,
  CONSTRAINT fk_OrderCustId FOREIGN KEY (ocCustomerId)
  REFERENCES tblCustomer(crCustomerId),
  CONSTRAINT fk_OrderEmpId FOREIGN KEY (ocEmpId)
  REFERENCES tblEmployee(eeEmpId)
);
```

```
CREATE TABLE tblStock
(
  skOrderCld INT,
  skWineOld INT,
  skNumCases INTEGER NOT NULL,
  CONSTRAINT fk_ScOrderId FOREIGN KEY (skOrderCld)
  REFERENCES tblOrderCust(ocOrderCld),
  CONSTRAINT fk_SwOrderId FOREIGN KEY (skWineOld)
  REFERENCES tblOrderWine(weWineOld)
);
```

DESIGN // Model Data

To test your database you will need to include some model data – 3 to 5 rows per table.

tblCustomer				
crCustomerId	crCompanyName	crAddress	crSuburb	crPhone
1	Clarion Hotel	326 Canterbury Rd	Forest Hill	9878 4111
2	Notting Hill Hotel	260 Ferntree Gully Rd	Notting Hill	9544 3031
3	The Prince Hotel	2 Acland St	St Kilda	9536 1111
4	Evelyn Hotel	351 Brunswick Street	Fitzroy	9419 5500
5	Bundoora Hotel	49 Plenty Rd	Bundoora	9468 9999
6	Napier Hotel	210 Napier St	Fitzroy	9419 4240

tblEmployee			
eeEmpId	eeEmpName	eeWineold	eeOrdercId
1	Paul Readings	6	5
2	Simon Fields	5	4
3	Bret Anderson	3	1
4	Sam Johnson	1	3
5	Matthew Long	2	2
6	James Jolly	4	6

tblOrderCust					
ocOrdercId	ocOrderDate	ocTotalCost	ocPaid	ocDatePaid	ocCustomerId
1	29/08/2008	\$	Y	01/09/2008	1
2	01/09/2008	\$	Y	03/09/2008	2
3	10/09/2008	\$	Y	12/09/2008	3

4	11/09/2008	\$	Y	17/09/2008	4
5	15/09/2008	\$	N		5
6	17/09/2008	\$	N		6

tblOWStatus

osStatusId	osStatus	osOWDate	osWineold
1	YES	03/09/2008	1
2	YES	11/09/2008	2
3	YES	15/09/2008	3
4	YES	15/09/2008	4
5	YES	15/09/2008	5
6	YES	17/09/2008	6

tblStock

skOrdercid	skWineold	skNumCases
1	5	25
2	4	15
3	2	40
4	3	28
5	1	30
6	6	22

tblSupplier						
srSupplierId	srSupplierName	srAddress	srSuburb	srState	srPCode	srPNumber
1	Wolf Blass	97 Sturt Highway	Nuriootpa	SA	5355	8568 7311
2	Mitchelton	Mitchellstown Rd	Nagambie	VIC	3608	5736 2222
3	Brown Brothers	5 Whitehorse Rd	Balwyn	VIC	3175	9817 1234
4	Tyrrell's	5/ 51- 55 City Rd	Southbank	VIC	3006	9686 3397
5	McWilliams	8 Ferntree Gully Rd	Scoresby	VIC	3179	1800 800 584
6	De Bortoli	6 -8 Research Drv	Croydon South	VIC	3136	9761 4100

tblOrderWine						
weWineold	1	2	3	4	5	6
weWineClass	Table Wine	Table Wine	Table Wine	Table Wine	Table Wine	Table Wine
weProducer	Wolf Blass	Mitchelton	Brown Brothers	Tyrrell's	McWilliams	De Bortoli
weLabelTitle	Gold Label Riesling	Cresnet Shiraz	Shiraz	Futures Shiraz	Lilydale Vineyards Chardonnay	Hunter Valley Chardonnay
weWineColour	White	Red	Red	Red	White	White
weWineType	Riesling	Shiraz	Shiraz	Shiraz	Chardonnay	Chardonnay
weVintage	2000	2002	2001	2002	2001	2002
weCaseCost	\$130.50	\$80.95	\$140.50	\$80.50	\$100.00	\$97.60
weSupplierId	1	2	3	4	5	6

SELECT STATEMENTS // Test Queries

List the company name, the order date, the number of cases ordered, and the name of the wine.

```
SELECT ocCompanyName, ocOrderDate, skNumCases, weLabelTitle
FROM tblCustomer, tblOrderCust,
WHERE
```

List the employee names, the wine name and the number of cases the staff members have ordered.

```
SELECT a. eeEmpName, b.weLabelTitle, c.skNumCases
FROM tblEmployee as a, tblOrderWine as b, tblStock as c
WHERE a. eeWineold = b.weWineold
AND b.weWinold = c.skWinold;
```

SELECT STATEMENTS // (continued)

Insert Query – Insert new supplier details into supplier table

```
INSERT INTO tblSupplier (srSupplierId, srSupplierName, srAddress, srSuburb, srState, srPCode, srPNumber)
VALUES (7, 'Yellowglen', 'Whites Rd', 'Smythesdale', 'VIC', '3351', '5342 8617');
```

Update Query – Create an update for company name using primary key

```
UPDATE tblCustomer
SET crCompanyName = 'The Evelyn Hotel'
WHERE crCustomerId = 4;
```

Delete Query – Create a query to delete an employee from the database

```
DELETE tblEmployee
WHERE eeEmpId = 6;
```