Assignments for Relational Database Management System

Author(s)	Monika Verma, Satyendra Singh, Diwakar Jaiswal, Lakshmi D.L., Manjeet Singh Juneja
Customization for Campus Connect FP 4.1	Hari S.
Authorized By	Satheesha B Nanjappa
Creation/Revision Date	June 2017
Version	4.1

Infosys® | Building Tomorrow's Enterprise

Education, Training and Assessment
Infosys Limited
Electronics City
Hosur Road
Bangalore - 561 229, India.

Tel: 91 80 852 0261-270
Fax: 91 80 852 0362
www.infosys.com
mailto:ETA@infosys.com

Document Revision History

Version	Date	Author(s)	Reviewer(s)	Description
1.0	Jan 2014	Monika Verma, Satyendra Singh, Diwakar Jaiswal	Kiran R.K., Lakshmi D.L.	Initial Draft
1.1	Feb 2014	Monika Verma, Lakshmi D.L.	Kiran R.K.	As per feedback comments
2.0	May 2014	Monika Verma, Manjeet Singh Juneja	Kiran R.K.	Incorporated additional assignments

Customization Revision History

Version	Date	Author(s)	Reviewer(s)	Description
4.0	Sep 2014	Hari S.	Vani K.N.	Customization for Campus Connect FP 4.0 release
4.1	June 2017	Hari_S	Vani K.N.	Customization for Campus Connect FP 4.1 release

CONTENTS

COPYRIGHT NOTICE i

Document Revision History ii

Customization Revision History ii

CONTENTS iii

Assignments on Database basics 1

Assignment 1: Relational model: Keys - Guided activity 1

Assignment 2: Identifying data items – Guided activity 2

Assignment 3: ER modeling – Terms and notations – Guided activity 4

Assignment 4: ER model – Guided activity 6

Assignment 5: Conversion ER model to Relational schema - Guided activity 7

Assignment 6: Functional dependency – Guided activity 8

Assignment 7: Identify highest normal form – Guided activity 8

Assignments on SQL basics 11

Assignment 1: CREATE and INSERT - Demo 11

Assignment 2: DDL - CREATE table - Demo 13

Assignment 3: DML - Demo 18

Assignment 4: DDL – ALTER table - Demo 19

Assignment 5: DEFAULT – Guided Activity 22

Assignment 6: SELECT - Demo 23

Assignment 7: DML – Guided activity 25

Assignment 8: DISTINCT, ORDER BY - Demo 26

Assignment 9: DISTINCT, ORDER BY – Guided activity 27

Assignment 10: CASE statement - Demo 28

Assignment 11: CASE statement – Guided Activity 29

Assignment 12: SQL functions - Demo 30

Assignment 13: SQL functions – Guided activity 33

Assignments on GROUP BY and HAVING 35

Assignment 1: GROUP BY and HAVING – Guided activity 35

Assignments on Joins 35

Assignment 1a: INNER JOIN – Guided activity 35

Assignment 1b: INNER JOIN – Guided activity 37

Assignment 2a: LEFT OUTER JOIN – Guided activity 39

Assignment 2b: LEFT OUTER JOIN – Guided activity 39

Assignment 3: RIGHT OUTER JOIN – Guided activity 39

Assignment 4: Self-Join – Guided activity 40
Assignments on Sub queries 40
Assignment 1: Independent sub query - Single row and Multi row – Guided activity 40
Assignment 2: Multiple columns sub query - Demo 42
Assignment 3: Multiple columns sub query - Guided activity 43
Assignment 4: Correlated sub query – Guided activity 44
Assignment 5: Exists/Not Exists – Guided activity 45
Assignments on Index 45
Assignment 1: Index – Guided activity 45
Assignments on SQL Best practices 46
Assignment 1: Best practice tips for writing SQL queries 46
Assignments on Views 49
Assignment 1: Views - Demo 49
Assignment 2: Views – Guided activity 49
Assignments on Transaction and Locks 50
Assignment 1: Concurrency issues – Self Study 50
Assignment 2: Locks – Guided activity 53
Assignments on PL/SQL 54
Assignment 1: PL/SQL block – Guided activity 54
Assignment 2: Usage of BOOLEAN data type – Guided activity 56
Assignment 3: Usage of SQL% Attributes in PL/SQL – Guided activity 57

Assignments on Database Basics

Assignment 1: Relational model: Keys - Guided activity

Objective: Identify the candidate key, primary key and foreign keys for the given relations

Problem description: All the details pertaining to retail application are stored in the form of relations.

Note: Refer *CCFP4.1-RDBMS-EasyShop Retail Application Case Study.docx*

Few of the relations are mentioned below:

Relation Name	Description of relation
item	items being sold in the retail store
supplier	suppliers from whom items are bought to the retail store
quotation	quotations placed by the suppliers
orderstatus	order status against the accepted quotations
retailoutlet	details of the various retail outlets
employee	employees working for the retail outlet
customer	customers who buy items from retail outlets
inwarditem	quantity of items procured to warehouse against orderid
outwarditem	items shipped from warehouse to retail outlet

The relations along with their attributes are mentioned below. Identify the candidate key, primary key and foreign key(s) for these relations:

a) **item** (itemcode, itemtype, description, price, reorderlevel, quantityonhand, category)

b) **supplier** (supplierid, suppliername, suppliercontactno, supplieremailid)

c) **quotation** (quotationid, supplierid, itemcode, quotedprice, quotationdate, quotationstatus)

d) **orderstatus** (orderid, quotationid, orderdate, status, paymentdate, amountpaid, paymentmode)
e) **retailoutlet** (retailoutletid, retailoutletlocation, retailoutletmanagerid)
f) **employee** (empid, empname, designation, emailid, contactno, worksin)
g) **customer** (customerid, customertype, customername, emailid, contactno, address)
h) **inwarditem** (orderid, inwardqty, inwarddate)
i) **outwarditem**(outwardid, itemcode, retailoutletid, outwardqty, shipmentdate)

Estimated time: 60 mins

Summary of this assignment: Identification of candidate key, primary key and foreign key

Assignment 2: Identifying data items – Guided activity

Objective: Given the case study, identify the data items required

Problem description:

1. Consider the EasyShop application. Identify all the data items required for creating the database.
   Refer CCFP4.1-RDBMS-EasyShop Retail Application Case Study.docx

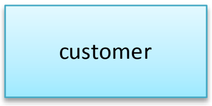
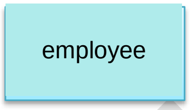
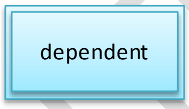
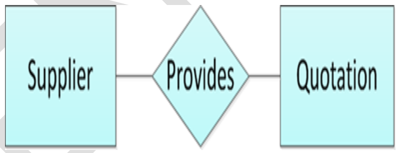
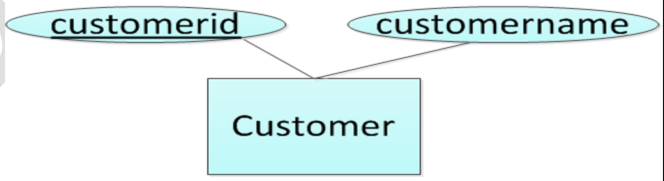
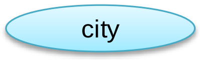
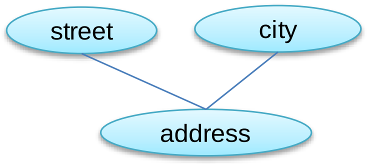
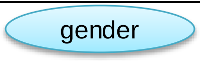
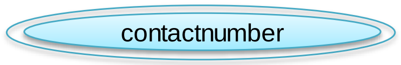
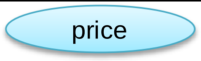
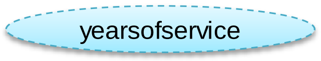
2. In a training institute, the results of the assessments conducted for various courses needs to be stored. An application is used which has the database to store the marks details. Each assessment is made up of two or more components.
   The senior management wants to analyze the scores of the participants. Reports need to be generated so that analysis can be made – course-wise, topic-wise, college-wise and region-wise. Identify the data items that are required for the database.

Estimated time: 20 minutes

Summary of this assignment: Identification of the data items for a given case study

Assignment 3: ER modeling – Terms and notations – Guided activity

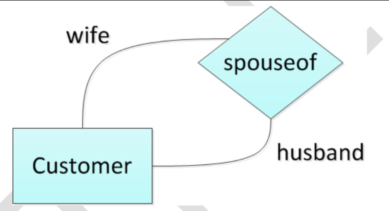
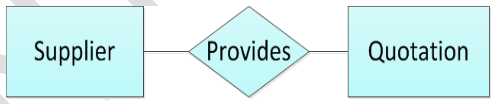
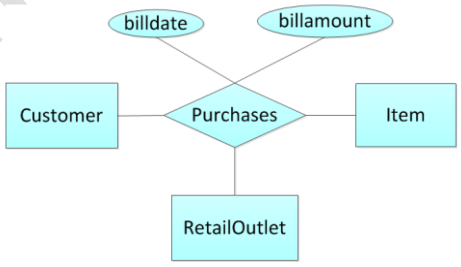
Problem description: Terms in ER modeling with examples and notation

Term	Definition	Example and Notation
Entity	An entity is a real world object that has an independent existence, about which we intend to collect data. It need not be a material existence	
Strong entity	Strong entity: Entity that has its own key attribute(s)	
Weak entity	Weak entity: The entity for which the existence is dependent on other entity	
Relationship	It is defined as the association between two or more entities	
Attribute	Properties/Characteristics that describe entity	
Simple attribute	Simple: Can attribute cannot be divided into simpler components	
Composite attribute	Composite: Can split into components	
Single valued attribute	Single valued : Can take on only a single value for each entity instance	
Multi-valued attribute	Multi-valued: Can take up many values	
Stored attribute	Stored Attribute :Attribute that needs to be stored permanently	
Derived attribute	Derived Attribute: Attribute that can be calculated based on other attributes.	

Note: Weak entities and the relationship involving weak entities represented via *Double Rectangle* and *Double Diamond,* respectively

Degree of relationship:

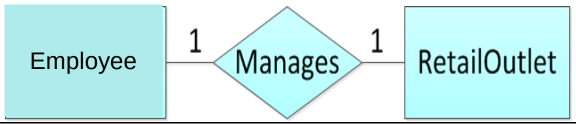
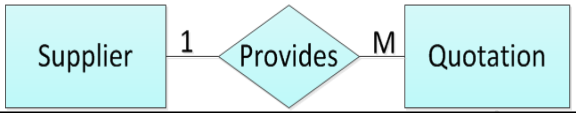
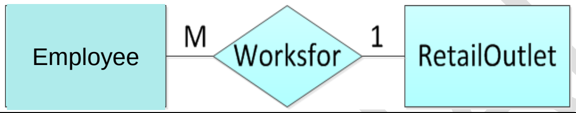
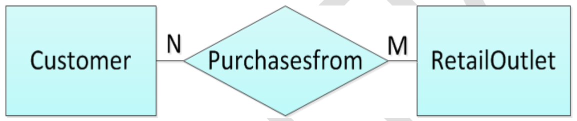
Number of entities involved in the relationship

Degree of relationship	Definition	Example
Unary	Relationship involving only one entity	
Binary	Relationship involving two entities	
Ternary	Relationship involving three entities	 billdate billamount Customer Purchases Item RetailOutlet

Cardinality of relationship:

The number of instances of one entity is related to the number of instances of another entity.

Cardinality is represented by placing the appropriate number along with the relationship

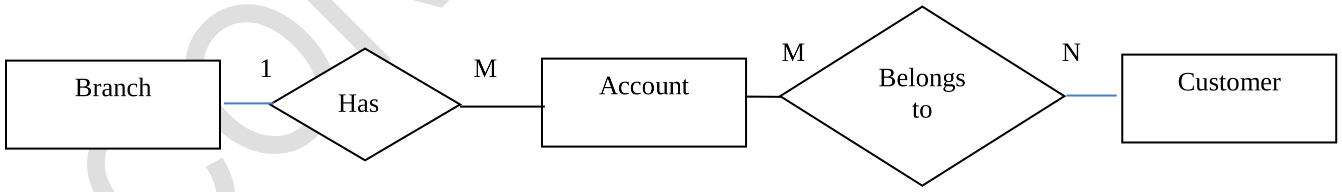
Cardinality of relationship	Example
one to one (1:1)	
one to many (1:M)	
many to one (M:1)	
many to many (M:N)	

Estimate time: 20 mins

Note: ERD is usually viewed from left to right and from top to bottom.

Assignment 4: ER model – Guided activity

Problem description: Consider the following ER-Diagram depicting a banking scenario and answer the questions below:

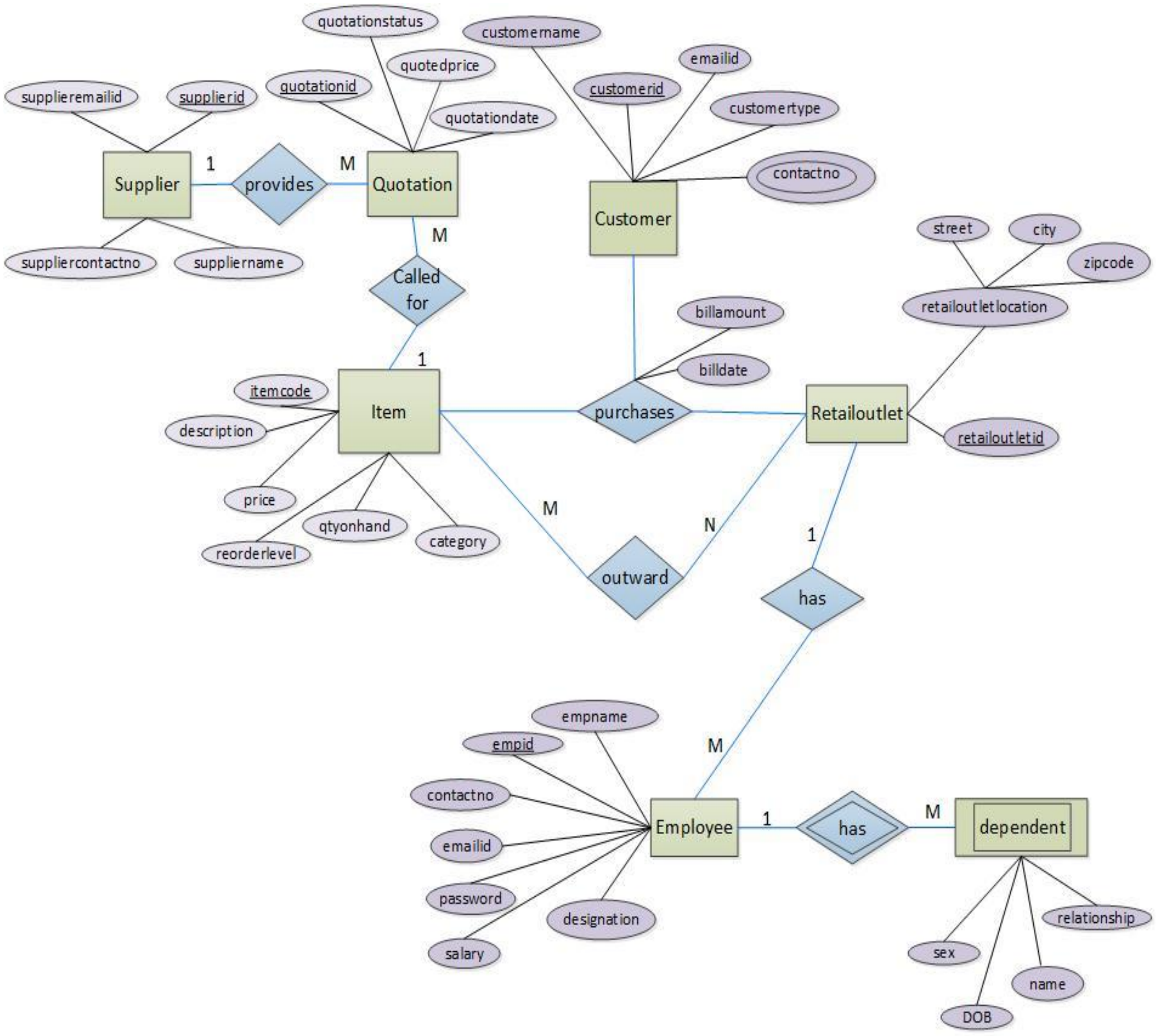


1. How many entities are present? Name those entities.
2. How many relationships are present? Name those relationships.
3. Mention the degree of all identified relationships.
4. Cardinality of all identified relationships.

Estimated time: 10 mins

Assignment 5: Conversion ER model to Relational schema - Guided activity

Problem description: Convert the below ER model to relational schema



1. Identify the strong and weak entities.
2. Identify the key attributes.
3. How many tables would be resulted after conversion to relational schema?
4. How many primary and foreign keys are there in resultant relational schema?

Estimated time: 45 mins

Summary of this assignment: Converting the ER model to relational schema

Assignment 6: Functional dependency – Guided activity

Objective: To determine the candidate key and various types of functional dependencies

Problem Description:

1. Consider the following relation and the given functional dependencies, identify the candidate key and type of functional dependency

   Relation:

   item (itemid, marketingid, vendor, style, price)

   Functional dependencies:

{itemid, marketingid}	→	price
itemid	→	vendor, style

2. Consider the following set of relations and the given functional dependencies, identify the candidate keys and type of functional dependency

   Relations:

   sale (customerid, product, salesrep)

   product (product#, vendor, vendorcity)

   Functional dependencies:

{customerid, product}	→	salesrep
product#	→	vendor
vendor	→	vendorcity

Estimated time: 30 mins

Assignment 7: Identify highest normal form – Guided activity

Objective: To determine the highest normal form

Problem Description:

1. Find out the candidate key(s) and the highest normal form in the given relation:

trainee (traineeid, traineename, classroomid, pcid)

Consider the following functional dependencies:

i.	traineeid	→	traineename
ii.	{classroomid, pcid}	→	traineename
iii.	traineeid	→	classroomid, pcid
iv.	{classroomid, pcid}	→	traineeid

Suggested Solution:

Step 1: Identify the candidate key, with the help of given functional dependencies.

As per the first functional dependency, let us assume traineeid is a candidate key. To confirm that, traineeid is a candidate key, it should determine rest of the attributes.

The given two functional dependencies shows that, we can determine rest of the attributes (i.e. traineename, classroomid and pcid} with the help of traineeid.

traineeid → traineename

traineeid → classroomid, pcid

Hence, traineeid is a candidate key. But as we know, a relation can have more than one candidate key. Let's check, if we have another candidate key in the given relation.

As per the second functional dependency, let us assume {classroomid, pcid} is a composite candidate key. Again if, {classroomid, pcid} is a composite candidate key, it should determine {traineename and traineeid}.

The given two functional dependencies illustrates that, we can determine {traineename and traineeid} with the help of {classroomid, pcid}.

{classroomid, pcid} → traineename

{classroomid, pcid} → traineeid

Hence, {classroomid, pcid} is a composite candidate key in a given relation.
But, to prove that it is a candidate key, the subset of {classroomid, pcid} should not determine rest of the attributes independently. In a given relation, it is not possible.

Hence, we have two candidate keys:

a) traineeid
b) {classroomid, pcid}

Step 2: Identify the highest normal form

a) Identify the key attributes and non-key attributes.
   As we know, attributes which are the part of candidate key is/are called key attribute(s) and others is/are non-key attributes. In our case,
   Key attributes are { traineeid, classroomid and pcid }
   Non-key attribute is { traineename }

b) Check for 1NF:
   As per the definition of 1NF, all the attributes should be atomic in nature.
   So, it holds for a given trainee relation.

c) Check for 2NF:
   As per the definition of 2NF, there should not be any partial dependency between non-key and key-attributes.
   It means we should not have a dependency like:
   Key-attribute (subset of composite candidate key) → Non-key attribute(s)

   In our case, the composite candidate key is {classroomid and pcid}.
   Let's check, do we have any dependency like;

classroomid	→	Non-key attribute
pcid	→	Non-key attribute

After looking at the given functional dependencies, we can conclude that, there is no partial dependency existing. Hence the given relation is in 2NF.

d) Check for 3NF:
Again, as per the definition of 3NF, relation should not have any transitive dependency between key and non-key attributes via another non-key attribute.
In simple words, we should not have any dependency among non-key attributes If Non-key-attribute → Non-key-attribute, then there is a transitive dependency and the relation will not be in 3NF.

In our case, we have only one non-key attribute, so there is no chance of having transitive dependency in a relation. Hence, our trainee relation is in 3NF.

Estimated time: 50 minutes

Assignments on SQL basics

Assignment 1: CREATE and INSERT - Demo

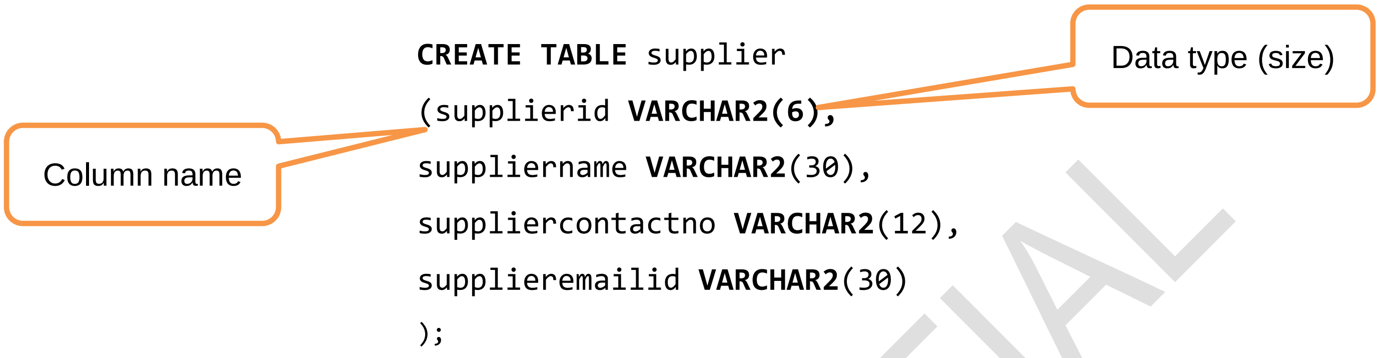
Objective: To create a table based on a business requirement/use case

Problem description:

1. For storing the supplier details, the supplier table needs to be created. Table is created without any constraints

Column name	Data type	Constraint	Description
supplierid	VARCHAR2(6)	None	ID of the supplier

suppliername	VARCHAR2(30)		Name of the supplier
suppliercontactno	VARCHAR2(12)		Contact number of supplier
supplieremailid	VARCHAR2(30)		Email id of supplier



```
CREATE TABLE supplier
(supplierid VARCHAR2(6),
suppliername VARCHAR2(30),
suppliercontactno VARCHAR2(12),
supplieremailid VARCHAR2(30)
);
```

2. Insert a record in the supplier table

```
INSERT INTO supplier
(supplierid, suppliername, suppliercontactno,
supplieremailid)
VALUES
('S1001', 'Giant Store', '203-237-2079',
'rachel1@easy.com');
```

Note:

1) SQL is case insensitive language.
2) Coding convention:
    a. All the keywords must be written in UPPER case and table name, column name etc. must be written in lower case.
3) The values of the columns whose data type is CHAR, VARCHAR2 or DATE must be mentioned in single quotes.
   Example: `'S1001', '15-Aug-1947' etc.`
4) The values that are mentioned after the VALUES keyword must follow the same order as the columns mentioned after the table name
5) The **data values** for VARCHAR2 or CHAR data type **is case sensitive**

3. SQL statement to add a record into the “supplier” table for a supplier without email id is as follows

```
INSERT INTO supplier (supplierid, suppliername,
suppliercontactno, supplieremailid)
VALUES ('S1002','EBATs','115-340-2345', NULL);
```

4. SQL to retrieve the details of suppliers

```
SELECT * FROM supplier;
```

Note: More details about SELECT statement will be dealt later

Estimated time: 20 mins

Summary of this assignment: You have learnt to create table without any constraints and insert a record into the table.

Assignment 2: DDL - CREATE table - Demo

Problem description:

1. Create a table supplier where supplier id should not be repeated

   Step 1: Existing table to be dropped by using below DDL statement

   ```
   DROP TABLE supplier;
   ```

   Step 2: Create supplier table, with “supplierid“ column to have UNIQUE values

   ```
   CREATE TABLE supplier(
   supplierid VARCHAR2(6)
   CONSTRAINT sup_sid_unq UNIQUE,
   suppliername VARCHAR2(30),
   suppliercontactno VARCHAR2(12),
   supplieremailid VARCHAR2(30)
   ```

```
            );
```

Best practice:

1) Every constraint must be named
2) The naming convention to be followed for a constraint:

   `<short table name>_<short column name>_<short constraint type>`
   Example: **`sup_sid_unq`**

Try-out and observe the output :

1) Insert two records into supplier table having same supplier id
2) Insert two records into supplier table with supplier id as NULL value

2. Create the table supplier where supplier id must not be left blank and it should not repeat too

```
        DROP TABLE supplier;

        CREATE TABLE supplier (
        supplierid VARCHAR2(6)
        CONSTRAINT sup_sid_unq UNIQUE
        CONSTRAINT sup_sid_nn NOT NULL,
        suppliername VARCHAR2(30),
        suppliercontactno VARCHAR2(12),
        supplieremailid VARCHAR2(30)
    );
```

A column can have more than one constraint

Repeat the 2 inserts of previous try-out and observe the output

3. To create a table supplier with supplier id as primary key

```
DROP TABLE supplier;

CREATE TABLE supplier(
supplierid VARCHAR2(6)
CONSTRAINT sup_sid_pk PRIMARY KEY,
suppliername VARCHAR2(30),
suppliercontactno VARCHAR2(12),
supplieremailid VARCHAR2(30)
);
```

4. Create the table retailstock with retailoutletid and itemcode as **composite primary key.**

Column name	**Data type**	Constraint	**Description**
retailoutletid	VARCHAR2(6)	Primary key	ID of the supplier
itemcode	VARCHAR2(6)		Item code
retailunitprice	NUMBER		Unit price of item
quantityavailable	NUMBER		Quantity of items available in the retail outlet

```
CREATE TABLE retailstock (
retailoutletid VARCHAR2(6),
itemcode VARCHAR2(6),
retailunitprice NUMBER,
quantityavailable NUMBER,
CONSTRAINT rtlstk_rid_icode_pk PRIMARY
KEY(retailoutletid, itemcode)
);
```

Note:

1) The composite primary key must be created as a table level constraint

Try-out and observe the output :

Insert a record into retailstock table with:

1) Retailoutletid as ‘R1001’ and Itemcode as ‘I1001’.
2) Retailoutletid as ‘R1001’ and Itemcode as ‘I1002’.
3) Retailoutletid as ‘R1001’ and Itemcode as ‘I1001’.
4) Retailoutletid as ‘R1002’ and Itemcode as ‘I1001’.
5) Retailoutletid as NULL and itemcode as ‘I1001’.

5. Create table supplier where the supplier id must start with ‘S’

```
DROP TABLE supplier;

CREATE TABLE supplier
(supplierid VARCHAR2(6)
CONSTRAINT supl_sid_pk PRIMARY KEY
CONSTRAINT supl_sid_ck CHECK (supplierid LIKE 'S%'),
suppliername VARCHAR2(30),
suppliercontactno VARCHAR2(12),
supplieremailid VARCHAR2(30));
```

Try-out and observe the output :

1) Insert a record into supplier table with supplierid as '1001' and observe the output.
2) Insert a record into supplier table with supplierid as 'S1001' and observe the output.

6. Create quotation table with supplierid as foreign key:

Column name	Data type	Constraint	Description
quotationid	VARCHAR2(6)	Primary key	Unique ID
supplierid	VARCHAR2(6)	References supplier	ID of supplier
itemcode	VARCHAR2(10)		Item code
quotedprice	NUMBER		Price of the item
quotationdate	DATE		Date of quotation
quotationstatus	VARCHAR2(10)		Status of quotation

Question: What should be the values for supplierid column?

```
CREATE TABLE quotation (
quotationid VARCHAR2(6) CONSTRAINT quot_qid_pkey PRIMARY
KEY,
supplierid VARCHAR2(6)
CONSTRAINT quot_sid_fk REFERENCES supplier(supplierid),
itemcode VARCHAR2(10),
quotedprice NUMBER,
quotationdate DATE,
quotationstatus VARCHAR2(10)
);
```

Try-out and observe the output :
1) Insert a record into supplier table with supplierid as 'S1002' and insert a record into quotation table with supplierid as 'S1003'

2) Insert a record into supplier table with supplierid as ‘S1003’ and insert a record into quotation table with supplierid as ‘S1003’
3) Insert a record into quotation table with supplierid as NULL

Estimated time: 45 mins

Assignment 3: DML - Demo

Objective: To perform DML operations on a table based on given a relational schema representing a business requirement/use case.

Problem description:

1. Include a row in the supplier table

```
INSERT INTO supplier
VALUES ('S1009','M Stores','103-237-2017', 'alex@easy.com');
```

Note: The statement written above is different from the one mentioned in assignment 2

1) After the table name, the column names are not mentioned. Hence, the data values after the VALUES keyword must be in the same order as mentioned during table creation

2. The contact details of supplier ‘S1009’ has changed to 303-537-9127. Make the changes in the table

```
UPDATE supplier
SET suppliercontactno = ‘303-537-9127’
WHERE supplierid=‘S1009’;
```

3. The emailid of supplier ‘S1009’ is modified to ‘john@ebats.com’ and the contact number to ‘879-456-398’

```
UPDATE supplier
SET suppliercontactno = ‘879-456-398’,
supplieremailid = ‘john@ebats.com’
 WHERE supplierid=‘S1009’;
```

4. The supplier ‘S1009’ is no longer in business with EasyShop. The details of supplier ‘S1009’ needs to be removed

```
DELETE FROM supplier
WHERE supplierid=‘S1009’;
```

Estimated time: 15 mins

Assignment 4: DDL – ALTER table - Demo

Objective: To modify the structure of a table based on given relational schema representing a business requirement/use case.

Problem description:

1. After creating the table supplier, EasyShop wants to store the supplier city information also in the supplier table

```
ALTER TABLE supplier
ADD suppliercity VARCHAR2 (10);
```

2. After including the new supplier city column, EasyShop wants to increase the column size to 20

```
ALTER TABLE supplier
MODIFY suppliercity VARCHAR2 (20);
```

3. Now EasyShop would like to remove the supplier city details

```
ALTER TABLE supplier
DROP (suppliercity);
```

Note: Modification(downsize/change in data type) of a column is allowed only if the column is empty

At least one column must be present after dropping column/s in a table.

4. The structure of item table is as mentioned below. The “itemcode” column of “quotation” table created in assignment 3 is required to refer item table as a foreign key

Table: item

Column name	Data type	Constraint	Description
itemcode	VARCHAR2(6)	Primary key	Item code
itemtype	VARCHAR2(30)		Type of item
description	VARCHAR2(255)	Not null	Description of item
price	NUMBER(19,4)		Unit Price of item
category	CHAR(1)		Category of item

```
CREATE TABLE item(
itemcode VARCHAR2(6)
CONSTRAINT itm_icode_pk PRIMARY KEY,
itemtype VARCHAR2(30),
description VARCHAR2(255)
CONSTRAINT itm_desc_nn NOT NULL,
price NUMBER(10,4),
category CHAR(1)
);
```

SQL statement to modify the quotation table

```
ALTER TABLE quotation
ADD CONSTRAINT quot_icode_fk FOREIGN KEY (itemcode)
REFERENCES item (itemcode);
```

5. The category column of item table must have values, is a new requirement. To do this, add NOT NULL constraint on category column by executing the below SQL statement.

```
ALTER TABLE item
ADD CONSTRAINT itm_cat_nn NOT NULL (category);
```

Question: What is the output?

Solution:

```
ALTER TABLE item
MODIFY category CONSTRAINT itm_cg_nn NOT NULL;
```

Note:
1) To modify the data type of column, the column should be empty.
2) If required, to add a constraint on existing column of a table, the data present in that column must not violate the constraint that has to be added.
3) To change the column which can accept null values to not null, use MODIFY clause of ALTER statement.

What if the category column contains NULL value(s)?

Estimated time: 30 mins

Assignment 5: DEFAULT – Guided Activity

Note:
1. The default value specified will be used when no value is specified for the column while inserting data
2. The default value specified in the definition should satisfy the data type and length of the column
3. If a default value is not explicitly set, the default value for the column is implicitly set to NULL

Example:

```
CREATE TABLE quotation (
quotationid VARCHAR2(6) CONSTRAINT quot_qid_pkey PRIMARY
KEY,
quotedprice NUMBER,
quotationstatus VARCHAR2(10) DEFAULT 'Open'
CONSTRAINT quot_quotstatus_check
CHECK (quotationstatus IN
('Open','Accepted','Rejected','Closed'))
);
```

Inserting a row into the quotation table with the quotation status as a default value:

```
INSERT INTO quotation VALUES('Q1001',1000, DEFAULT);
```

Inserting a row into the quotation table with quotation status is other than default value:

```
INSERT INTO quotation VALUES('Q1002',1400, 'Accepted');
```

Estimated time: 20 mins

Assignment 6: SELECT - Demo

Objective: To perform SELECT operations on a table based on given a relational schema representing a business requirement/use case.

Note

- Refer the document *“CCFP4.1-RDBMS-EasyShop Retail Application_DB_Design.docx”* to know the DB design of EasyShop retail application.

Problem description:

1. Retrieve all the records of item table.

```
SELECT *
FROM item;
```

OR

```
SELECT itemcode, itemtype, description, price, category
FROM item;
```

2. Retrieve the supplier name and supplier’s contact number of the supplier ‘S1002’.

```
SELECT suppliername, suppliercontactno
FROM supplier
WHERE supplierid = 'S1002';
```

3. Retrieve the quotation id and supplier id of the quotations which have been either ‘Accepted’ or ‘Rejected’.

```
SELECT quotationid, supplierid
FROM quotation
WHERE quotationstatus ='Accepted'
OR quotationstatus =' Rejected';
```

OR

```
SELECT quotationid, supplierid
FROM quotation
WHERE quotationstatus
IN ('Accepted', ‘Rejected');
```

Try out and observe the output:

Retrieve the quotation id and supplier id of the quotations which are ‘CLOSED’

4. Retrieve supplier details like supplier id and supplier name whose names have ‘i’ as the second character.

```
SELECT supplierid, suppliername
FROM supplier
WHERE suppliername LIKE '_i%';
```

5. Retrieve the order details like **orderid**, **quotationid**, **status**, **paymentdate** for those orders where payments are not received.

```
SELECT orderid, quotationid, status, paymentdate
FROM orderstatus
WHERE amountpaid IS NULL;
```

6. Retrieve the quotation details like **quotationid**, **quotationdate**, **quotedprice** for those quotations which are quoted in the range of 1400 and 2150.

```
SELECT quotationid, quotationdate, quotedprice
FROM quotation
WHERE quotedprice>=1400 AND quotedprice<=2150;
```

OR

```
SELECT quotationid, quotationdate, quotedprice
FROM quotation
WHERE quotedprice BETWEEN 1400 AND 2150;
```

7. The management of EasyShop wants to increase the salary of all employees by 10%. Write a query to display the employee details along with their increased salary.

```
SELECT empid, empname, designation, salary,
salary*1.10 “NewSal”
FROM employee;
```

Estimated time: 30 mins

Assignment 7: DML – Guided activity

Objective: To perform SELECT operations on a table based on given a relational schema representing a business requirement/use case.

Note:

The table creation scripts and the insertion scripts for the retail application are available in “*Easyshop_Retailapplication_Table_Creation_Script.txt*”. Execute the SQL statements in this file and create the tables along with data for practicing various SELECT statements

Problem description: Solve the following queries:

1. Display **description** and **price** of different sizes of all ‘Hard disk’.
2. Display **quotationid, supplierid, itemcode, quotedprice, quotationdate, quotationstatus** of those quotations which are not accepted.
3. Retrieve the **designation** and **salary** of all ‘Manager’ and ‘Billing Staff’
4. Retrieve the **designation** and **salary** of all ‘Manager’ and ‘Billing Staff’ who have salary in the range of 2500 to 5000 (both inclusive).
5. Retrieve the retailoutletid and retailoutletlocation which does not have a Manager.
6. Retrieve the **orderid**, **quotationid** and **orderstatus** of those quotations where order is placed between the dates ‘1-Dec-2010’ and ‘1-Jan-2011’
7. Increase the unit price of all apparels by 10%

Estimated time: 60 mins

Assignment 8: DISTINCT, ORDER BY - Demo

Objective: To perform SELECT operations with DISTINCT and ORDER BY clause based on given relational schema representing a business requirement/use case.

1. Retrieve the different item types.

```
SELECT DISTINCT itemtype
FROM item;
```

2. Retrieve the different item types and category of the items.

```
SELECT DISTINCT itemtype, category
FROM item;
```

3. Retrieve the **itemcode**, **description** and **price** with the increasing order of item's price.

```
SELECT itemcode, description, price
FROM item
ORDER BY price;
```

4. Retrieve the different item types and category of the items in the increasing order of item types and category.

```
SELECT DISTINCT itemtype, category
FROM item
ORDER BY itemtype, category DESC;
```

Estimated time: 20 mins

Assignment 9: DISTINCT, ORDER BY – Guided activity

1. Retrieve the **designation** and **salary** of employees without any duplication of data.

2. Retrieve the **empname**, **designation** and **salary** in increasing order of the salary.

3. Retrieve the **empname**, **designation** and **salary** in increasing order of the designation and salary.

4. Retrieve the **empname** and **salary** in increasing order of the designation and decreasing order of salary.

Estimated time: 40 mins

Assignment 10: CASE statement - Demo

Objective: Given a relational schema representing a set of requirements, be able to write query using CASE statement and retrieve results.

Background: Consider a scenario of EasyShop retail system managing several retail outlets, items, customers and business operations.
The required schema and data has already been created using "***Easyshop_RetailApplication_TableCreationScript.txt"*** script

Problem Description: Write the queries for the following requirements.

1. Salary hikes are being given to all employees of EasyShop based on their role. The percentage increase is as given below. Write a query to display the employee details along with the increase in their salary based on the following conditions.

Designation(Role)	Hike in %
Administrator	10
Manager	5
Billing Staff	20
Security	25
Others	2

```
SELECT empid, empname, designation, salary,
CASE designation
WHEN 'Administrator' THEN salary*1.1
WHEN 'Manager' THEN salary*1.05
WHEN 'Billing Staff' THEN salary*1.20
WHEN 'Security' THEN salary*1.25
ELSE salary*1.02 END newsal
FROM employee;
```

2. The management of EasyShop would like to classify the salary of employees as Class 3, Class 2 and Class 1. The classification is done as if salary is less than 2500 then the class is ‘Class 3’ , if between 2500 and 5000 then ‘Class 2’, and if salary is more than 5000 then 'Class1'. Write a query to display the same.

```
SELECT
empname, salary,
CASE
  WHEN salary < 2500 THEN 'Class 3'
  WHEN salary BETWEEN 2500 AND 5000 THEN 'Class 2'
  WHEN salary > 5000 THEN 'Class 1'
END as Salgrade
FROM employee;
```

Estimated time: 30 minutes

Summary of this assignment: In this assignment you have learnt the usage of case statements and solve queries on them.

Assignment 11: CASE statement – Guided Activity

Objective: Given a relational schema representing a business requirement and an identified set of requirements, be able to write query using CASE statement and retrieve results.

Background: Consider a scenario of EasyShop retail system managing several retail outlets, items, customers and business operations. The required schema and data has already been created using "***Easyshop_RetailApplication_TableCreationScript.txt***" script and answer the following queries:

3. Write a query to update the salaries to all employees based on the conditions given in the above assignment.

4. The management of EasyShop would like to classify the items as cheap, affordable, expensive and very expensive. The classification is done if item unit price is between 0 and 499 then 'Cheap', if between 500 and 1999 then 'Affordable', if between 2000 and 4999 then 'Expensive' and if price is more than or equal to 5000 then 'Very Expensive'. Write a query to display the same.

Estimated time: 30 minutes

Summary of this assignment: In this assignment you have learnt the usage of case statements and solve queries on them.

Assignment 12: SQL functions - Demo

Objective: To perform SELECT operations along with SQL functions on a table based on given a relational schema representing a business requirement/use case.

1. For a discount of 25.5% being offered on all FMCG item's unit price, display item code, existing unit price as "Old Price" and discounted price as "New Price". Round off the discounted price to two decimal values.

```
SELECT itemcode, price "Old Price",
ROUND (price*0.745, 2) "New Price"
FROM item
WHERE itemtype = 'FMCG';
```

2. Retrieve the employee id, employee name of billing staff and the retail outlet where they work. Perform a case insensitive search.

```
SELECT empid, empname, worksin
FROM employee
WHERE UPPER (designation) = UPPER ('Billing staff');
```

OR

```
SELECT empid, empname, worksin
FROM employee
WHERE UPPER (designation) = 'BILLING STAFF';
```

OR

```
SELECT empid, empname, worksin
FROM employee
WHERE LOWER (designation) = 'billing staff';
```

3. Retrieve the order id, order status and payment mode of all the orders. Display 'Payment yet not done' when payment mode has NULL value.

```
SELECT orderid, status,
NVL(paymentmode, 'Payment yet not done')
FROM orderstatus;
```

4. Retrieve the description of items which have more than 15 characters.

```
SELECT description
FROM item
WHERE LENGTH (description)>15;
```

5. Display numeric part of supplier id.

```
SELECT SUBSTR (supplierid, 2, 4)
FROM supplier;
```

6. Retrieve the order id and the number of days between order date and payment date for all orders.

```
SELECT orderid, ABS(orderdate - paymentdate)
FROM orderstatus;
```

7. Retrieve the order id and the number of months between order date and payment date for all orders.

```
SELECT orderid, ABS (MONTHS_BETWEEN (orderdate,
paymentdate))
FROM orderstatus;
```

8. Insert a record into quotation table with quotation date exactly as '16/2/2014'.

```
INSERT INTO quotation VALUES ('Q1020', 'S1001',
'I1003', 125, '16/2/2014', 'Open');
```

What went wrong?

```
INSERT INTO quotation VALUES ('Q1020', 'S1001',
'I1003', 125, TO_DATE('16/2/2014', 'DD/MM/YYYY'),
'Open');
```

9. Display current date and current date as ‘Mon/DD/ YYYY Day’

```
SELECT SYSDATE, TO_CHAR(SYSDATE, 'Mon/dd/yyyy, Day')
FROM DUAL;
```

10. Retrieve the maximum salary, minimum salary, total salary and average salary of employees.

```
SELECT MAX (salary), MIN (salary), SUM (salary),
AVG (salary)
FROM employee;
```

11. Retrieve the total number of items available in warehouse.

```
SELECT COUNT (*)
FROM item;
```

12. Retrieve the total number of orders made and the number of orders for which payment has been done.

```
SELECT COUNT(orderid), COUNT(paymentdate)
FROM orderstatus;
```

13. Retrieve the total number of different item types available.

```
SELECT COUNT (DISTINCT itemtype)
FROM item;
```

Estimated time: 30 mins

Assignment 13: SQL functions – Guided activity

Objective: To perform SELECT operations along with SQL functions on a table based on given a relational schema representing a business requirement/use case.

Problem Description: Write the SQL queries for the following problem statements.

Try out and observe the output:

```
SELECT COUNT (*)
FROM orderstatus;
```

```
SELECT COUNT (amountpaid)
 FROM orderstatus;
```

1. Retrieve the average **salary** paid to employees.
2. Retrieve the **orderid** and the **number of months** between orderdate and paymentdate for all orders where the number of months is more than one and the payment has been done.
3. The salary of managers has been increased by 20%. Retrieve the **empid**, existing salary as “**Current Salary**”, increased salary as “**New Salary**” and the difference of existing and increased salary as “**Incremented Amount**”. Round off the increased salary up to two decimal places.
4. Display the **itemcode** of those items where the difference of quantity on hand and reorder level is more than 50.
   Hint: use `ABS()` function

Try out and observe the output:

```
1) SELECT MAX(empname) FROM employee;

2) SELECT AVG(empname) FROM employee;

3) SELECT MIN(inwarddate) FROM inwarditem;
```

Estimated time: 30 mins

Assignments on GROUP BY and HAVING

Assignment 1: GROUP BY and HAVING – Guided activity

Objective: Given a relational schema representing a business requirement/use case, be able to write queries using GROUP BY and HAVING

Problem description: Write SQL statements for the following requirements:

1. Retrieve the **orderid** and the **number of times** for which inward quantity arrived for that order

2. Retrieve the **itemcode** and **average of quantity available** where the average of quantity available is less than 75.

3. Display **paymentmode**, and **total number of payments** for those payments which were paid before the year 2011 and total number of payments should be more than 1.

4. Display the month and number of quotations received in each month.
   Hint: Use `TO_CHAR(quotationdate, ‘Month’)`

Estimated Time: 40 minutes

Summary of this assignment: Usage of GROUP BY and HAVING

Assignments on Joins

Assignment 1a: INNER JOIN – Guided activity

Objective: To learn about usage of INNER JOIN concepts.

Problem description: Solve the following queries:

1. There is a requirement to display **itemcode**, **supplierid** and **suppliername** for the suppliers who have given the quotations.
    a. _______ Number of tables are required to satisfy the above requirement.
    b. Name the identified tables.
    c. What is the joining condition?
    d. Write a query to meet the above requirement.

2. Display the **customerid** and **customername** of those customers who are also suppliers.
    a. _______ Number of tables are required to satisfy the above requirement.
    b. Name the identified tables.
    c. What is the joining condition?
    d. Write a query to meet the above requirement.

3. Display **cutomername** and **billamount** for those customers who have shopped for more than 5000
    a. _______ Number of tables are required to satisfy the above requirement.
    b. Name the identified tables.
    c. What is the joining condition?
    d. Is there any other condition to be checked? If yes, what is that?
    e. Write a query to meet the above requirement.

4. The Manager of EasyShop would like to know those supplier details with the quoted price of those items for which quotations have been accepted. Display **supplierid**, **suppliername**, **itemcode** and **quotedprice** for the same.

    a. _______ Number of tables are required to satisfy the above requirement.

b. Name the identified tables.

c. What is the joining condition?

d. Is there any other condition to be checked? If yes, what is that?

e. Write a query to meet the above requirement.

Estimated Time: 45 minutes

Summary of this assignment: Usage of INNER JOIN for various given business scenarios.

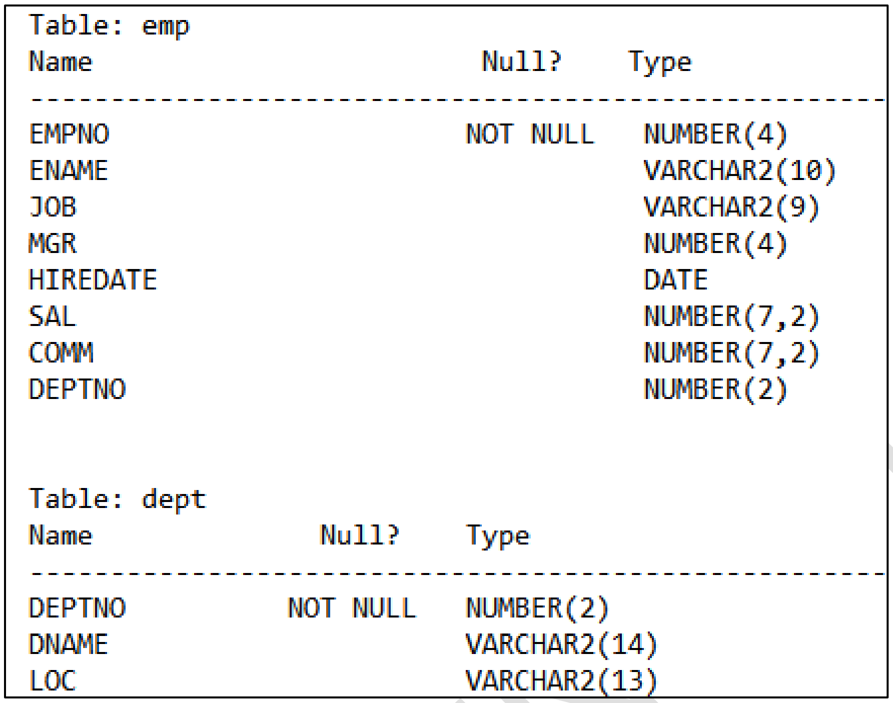
Assignment 1b: INNER JOIN – Guided activity

Objective: To learn about usage of INNER JOIN concepts.

Problem description: Solve the following queries:

Note: The table creation scripts and the insertion scripts for this assignment are available in "**Emp_Dept_Vehicle_Table_Creation_Script.txt**". Execute the SQL statements in this file and create the tables along with data for practicing various SELECT statements

1. Display name and salary of the employees, who are drawing salary more than 2000. Along with that display the department names in which they are working. Structure of the table is given below



```
Table: emp
Name                           Null?     Type
-----------------------------------------------------
EMPNO                          NOT NULL   NUMBER(4)
ENAME                                     VARCHAR2(10)
JOB                                       VARCHAR2(9)
MGR                                       NUMBER(4)
HIREDATE                                  DATE
SAL                                       NUMBER(7,2)
COMM                                      NUMBER(7,2)
DEPTNO                                    NUMBER(2)


Table: dept
Name                 Null?     Type
-----------------------------------------------------
DEPTNO          NOT NULL   NUMBER(2)
DNAME                      VARCHAR2(14)
LOC                        VARCHAR2(13)
```

Figure 1.1

2. Display the name of employees and their department names who are managers.

Note: See the structure in Figure 1.1

3. List the department names that have more than one employee drawing salary more than 1500, working under it.

Note: See the structure in Figure 1.1

Estimated Time: 30 minutes

Summary of this assignment: Usage of INNER JOIN for various given business scenarios.

Assignment 2a: LEFT OUTER JOIN – Guided activity

Objective: To learn about usage of LEFT OUTER JOIN concepts.

Problem description: Solve the following queries.

1. Display **itemcode**, **description** of all items of type 'FMCG' along with **outwardqty** and **retailoutletid** where the items have been moved.

2. For each item, identify the stock availability in the retail outlet R1001. Display **itemcode**, **description** and **qtyavailable** for the same. If there is no stock available for an item, display 'N.A.' for its quantity on available.

Estimated time: 20 minutes

Summary of this assignment: Usage of Left outer join

Assignment 2b: LEFT OUTER JOIN – Guided activity

Objective: To learn about usage of LEFT OUTER JOIN.

Problem description: Solve the following queries.

1. For each employee, identify the vehicle owned by them. Display **ename** and **vehicleid** for the same. Display name of employees even if they don't own any vehicle.

2. For each employee, identify the vehicle owned by them. Display **ename** and **vehiclename** for the same. Display name of employees even if they don't own any vehicle.

   Estimated time: 20 minutes

Assignment 3: RIGHT OUTER JOIN – Guided activity

Objective: To learn about usage of RIGHT OUTER JOIN.

Problem description: Solve the following query.

1. For every retail outlet, identify the employees working in it. Display **empid**, **empname**, **retailoutletid** and their **retailoutletlocation**.

Estimated time: 10 minutes

Assignment 4: Self-Join – Guided activity

Objective: To learn about usage of self-join.

Problem description: Solve the following queries:

1. Retrieve employee name, designation, and email id of those employees who work in the same retail outlet where George works. Do not display the record of George in the result.

2. Display the customer id and customer name of those customers who are co-located. Do not display the duplicate records/rows.

3. Retrieve the customer id, customer name, and customer type of those customers who are of the same customer type as that of customer id 2004. Do not display the record of customer id 2004 in the result.

Estimated time: 40 minutes

Assignments on Sub queries

Assignment 1: Independent sub query - Single row and Multi row – Guided activity

Objective: To learn about usage of sub-queries and be able to construct an independent sub query and retrieve results.

Background: Consider the scenario of EasyShop retail system managing several retail outlets, items, customers and business operations.

Note: The required schema and data has already been created using "***Easyshop_RetailApplication_TableCreationScript.txt*****"** script

Problem Description: Write the SQL queries for the following requirements.

1. Identify the items which are shipped from the warehouse to various retail outlets. Display **itemcode**, **itemtype**, **description** and **category** of those items.

2. Identify the item details that have the least quoted price with the quotation status as 'Rejected'. Display **itemcode**, **itemtype**, **description** and **category** of those items.

3. The management would like to know the details of the items which has maximum quoted price amongst the quotations that have status as 'Closed' or 'Rejected'. Display **itemcode** and **description** of those items.

4. Identify the item having second highest unit price. Display **itemcode, description** and **price** of those items.
5. Identify the supplier who has submitted quotation with the least quoted price amongst the quotations that have been accepted. Display **itemcode**, **description, supplierid** and **suppliername** for the same.

Estimated time: 40 minutes

Summary of this assignment: In this assignment you have learnt the concept and usage of independent sub queries.

Assignment 2: Multiple columns sub query - Demo

Objective: To learn about usage of sub-queries and be able to construct an independent sub query which involves multiple columns and retrieve results.

Background: Consider a scenario of EasyShop retail system managing several retail outlets, items, customers and business operations.

Note: The required schema and data has already been created using "***Easyshop_RetailApplication_TableCreationScript.txt***" script

Problem Description: The payroll department requires the details of those employees who have the highest salary in each designation. Write a query to retrieve the details of all these employees.

```
SELECT empid, empname, designation as "Role", salary
  FROM employee
  WHERE (designation, salary) IN
       (SELECT designation, MAX(salary)
              FROM employee group by designation);
```

Estimated time: 20 minutes

Summary of this assignment: In this assignment you have learnt the concept of sub queries returning multiple columns.

Assignment 3: Multiple columns sub query - Guided activity

Objective: To learn about usage of sub-queries and be able to construct an independent sub query which involves multiple columns and retrieve results.

Background: Consider a scenario of EasyShop retail system managing several retail outlets, items, customers and business operations.

Note: The required schema and data has already been created using "***Easyshop_RetailApplication_TableCreationScript.txt"*** script

Problem Description: Write the SQL queries for the following requirements.

1. Display the **retailoutletid**, **itemcode**, **description** and **outwardqty** of those items which are shipped to the same retail outlet id and in same quantity as that of Best Rice. Do not display the details of Best Rice in the output.

2. The management wants to have stock clearance sale by providing discounts on the costliest items in each item type. Display **retailoutletid**, **itemcode**, **itemtype**, **description** and **category** of those items having maximum retail unit price in each item type in each retail outlets.

Estimated time: 30 minutes

Summary of this assignment: In this assignment you have learnt the concept of sub queries returning multiple columns.

Assignment 4: Correlated sub query – Guided activity

Objective: Given a relational schema representing a set of requirements, be able to construct a correlated sub query and retrieve results.

Background: Consider a scenario of EasyShop retail system managing several retail outlets, items, customers and business operations.

Note: The required schema and data has already been created using **"Easyshop_RetailApplication_TableCreationScript.txt"** script

Problem Description: Write the SQL queries for the following requirements.

1. Display the **itemcode, description** and **quotationdate** for those items which are quoted below the maximum quotation price on the same day.

2. Warehouse supplies various items to the retail outlets as per the requirement. Identify the outward id in which the outward quantity is less than or equal to the average outward quantity of all the items supplied in the same retail outlet. Display **outwardid** and **itemcode** for the same.

3. Identify the supplier who has submitted the quotation for an item with the quoted price, less than the maximum quoted price submitted by all other suppliers, for the same item.
   Display **supplierid**, **suppliername** and **itemcode** for the identified supplier. Do not display duplicate records.

4. The payroll department requires the details of those employees who are getting the highest salary in each designation. Display **empid, empname, designation** and **salary** as per the given requirement.

Estimated time: 60 minutes

Summary of this assignment: In this assignment you have learnt the concept and usage of correlated sub queries.

Assignment 5: Exists/Not Exists – Guided activity

Objective: Given a relational schema representing a set of requirements, be able to construct a correlated sub query using EXISTS operator and retrieve results.

Problem Description: Write the SQL queries for the following requirements.

1. Display the customer id and customer name of those customers who have not purchased at all from any retail outlet.
2. Display the item code, description of those items which are shipped to any retail outlet.

Estimated time: 30 minutes.

Summary of this assignment: In this assignment you have learnt the concept and usage of exists and not exists.

Assignments on Index

Assignment 1: Index – Guided activity

Objective: Given a relational schema representing a business requirement and an identified set of requirements, be able to create an index and understand its usage.

1. -- Index on a single column

```
SQL> CREATE INDEX supplier_idx
       ON supplier(suppliercontactno);
```

2. -- Index on multiple column

```
            SQL> CREATE INDEX rs_itid_rup_idx
                 ON retailstock (itemcode, retailunitprice);
```

3. -- Drop an Index

```
             SQL> DROP INDEX supplier_idx;
```

4. -- Unique Index on a column

```
             SQL> CREATE UNIQUE INDEX supplier_idx
                  ON supplier(suppliercontactno);
```

Try-out and observe the output

1) Create an index on itemcode column of the item table.
2) Create a unique index on itemtype column of the item table.
3) Create an index on itemcode column of the quotation table.

Estimated time: 15 minutes

Summary of this assignment: In this assignment you have learnt the concept and usage of Index.

Assignments on SQL Best practices

Assignment 1: Best practice tips for writing SQL queries

Objective: To write SQL queries by using the industry best practices and formulating logically accurate queries

1. Statements should be indented so that individual lines are neatly aligned.
2. Every table in the FROM list should be on a new line.
3. Oracle keywords should be entered in upper case

4. Constants, program variables etc. should be on the right hand side of a WHERE or HAVING clause.

```
SELECT ename
    FROM emp
            WHERE empno = 7946;
```

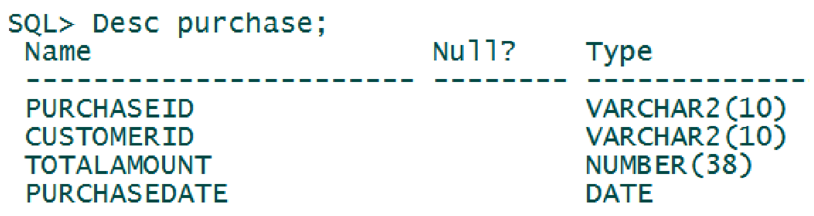
5. Do not use HAVING for columns that can be used in the WHERE clause. For example, do not write the following:

```
SELECT deptno, SUM(sal)
       FROM emp
            GROUP BY deptno
                   HAVING deptno IN(10, 20);
```

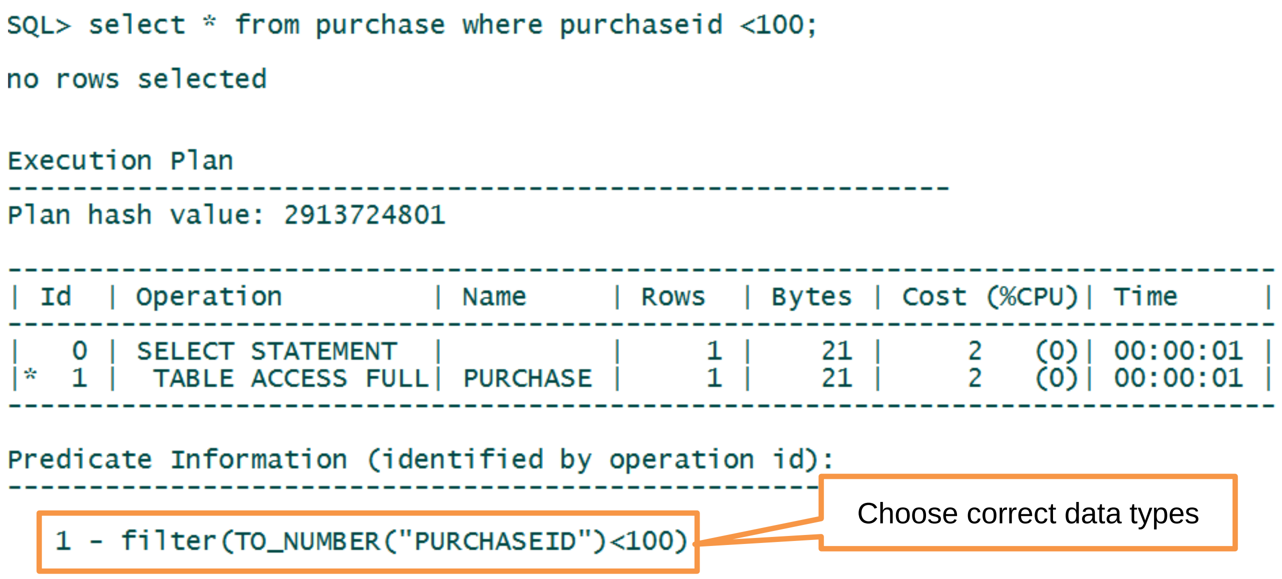
Optimized query

```
SELECT deptno, SUM(sal)
     FROM emp
           WHERE deptno IN(10, 20)
                    GROUP BY deptno;
```

6. Table aliases should be used in all queries that have more than one table in the FROM clause. The use of table aliases speeds up the parse phase of an oracle query, by reducing the number of recursive SQL queries.
7. When joining tables, the table returning the least number of rows should be last in the FROM list.
8. Consider the following structure of the purchase table. Write a query to retrieve purchaseid having value less than 100;



```
SQL> Desc purchase;
 Name                     Null?    Type
 ------------------------ -------- -------------
 PURCHASEID                        VARCHAR2(10)
 CUSTOMERID                        VARCHAR2(10)
 TOTALAMOUNT                       NUMBER(38)
 PURCHASEDATE                      DATE
```



```
SQL> select * from purchase where purchaseid <100;

no rows selected

Execution Plan
----------------------------------------------------------
Plan hash value: 2913724801

------------------------------------------------------------------------------
| Id  | Operation          | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |          |     1 |    21 |     2   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| PURCHASE |     1 |    21 |     2   (0)| 00:00:01 |
------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(TO_NUMBER("PURCHASEID")<100)
```

9. Remove unnecessary large-table full table scans. If the query returns less and 40 percent of the table rows in an ordered table or 7 percent of the rows in an unordered table), the query can be tuned to use an index in lieu of the full table scan.

10. Try to avoid the use of DISTINCT clause, where ever possible. As the DISTINCT clause will result in performance degradation, we should use this clause only when it is necessary or unavoidable.

11. Try to drop indexes that are not being used. Because each index takes up disk space and slow the DML operations, we should drop indexes that are not used.

Estimated time: 30 minutes

Summary of this assignment: In this assignment you have learnt a few tips for writing good queries

Assignments on Views

Assignment 1: Views - Demo

Objective: Given a relational schema representing a business requirement and an identified set of requirements, be able to create or query a view.

Problem Description: Create a view for displaying **empid**, **empname**, **designation** and **salary** details of all employees.

```
SQL> CREATE OR REPLACE VIEW emp_vw1 AS
SELECT empid, empname, designation, salary
FROM employee;

SQL> SELECT * FROM emp_vw1;

--Views with column alias

SQL> CREATE OR REPLACE VIEW emp_vw2(employeeid,
employeename, role, salary) AS
SELECT empid, empname, designation, salary
FROM employee;

SQL> SELECT * FROM emp_vw2;
```

Estimated time: 10 minutes

Assignment 2: Views – Guided activity

Objective: Given a relational schema representing a business requirement and an identified set of requirements, be able to create or query a view.

Problem Description:

1. Insert sample record with designation as ‘Manager’ into emp_vw1 view which is created in the previous assignment and verify for the inserted record in the base table.
2. Increase the salary by 10% of all employees with designation as ‘Manager’ using the view emp_vw1 and verify for the updated record in the base table.

3. Remove the details of employees added in the problem description 1 of this assignment using the view emp_vw1 and verify for the deleted record in the base table.

Estimated time: 30 minutes

Summary of this assignment: In this assignment you have learnt the concept and usage of Views

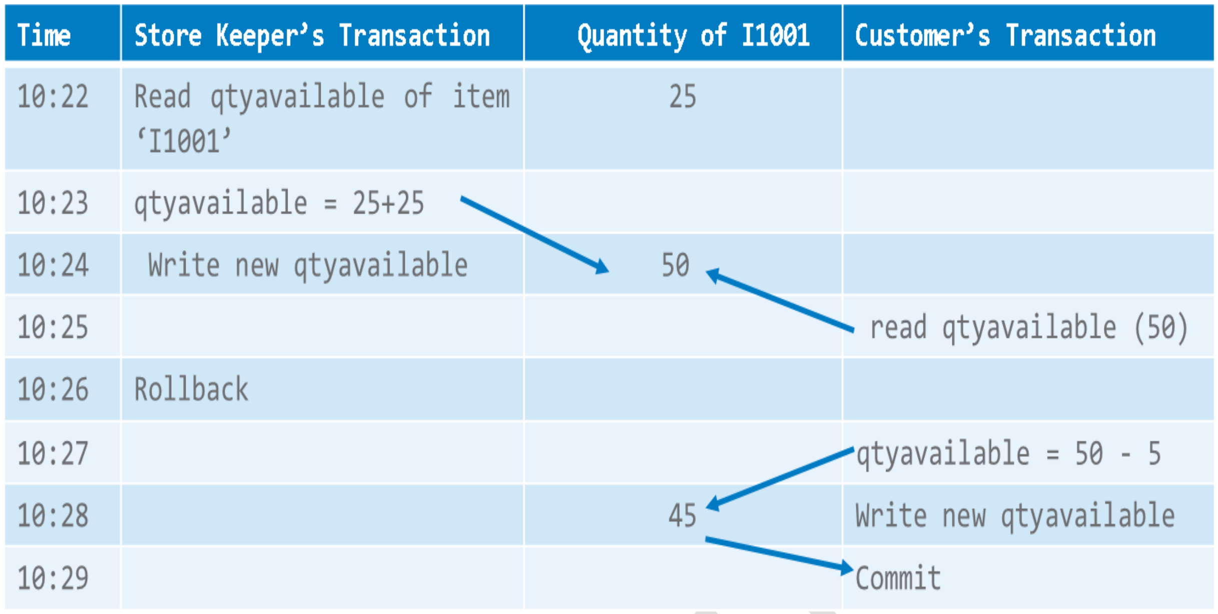
Assignments on Transaction and Locks

Assignment 1: Concurrency issues – Self Study

Objective: To understand concurrency issues

Problem description:

1. A store keeper is adding 25 units of item I1001 in retail outlet R1001. But then he realized that he has to update the item I1002 and not I1001. So he undid the update operation. Simultaneously 5 units of item I1001 is being purchased from retail outlet R1001 by a customer. Sequence of the transactions are as shown below:
   What is your observation?



Time	Store Keeper's Transaction	Quantity of I1001	Customer's Transaction
10:22	Read qtyavailable of item 'I1001'	25	
10:23	qtyavailable = 25+25		
10:24	Write new qtyavailable	50	
10:25			read qtyavailable (50)
10:26	Rollback		
10:27			qtyavailable = 50 - 5
10:28		45	Write new qtyavailable
10:29			Commit

Dirty read

How many items should be present in the outlet finally? How can this be resolved?

2. A store keeper is moving 25 units of item I1001 from warehouse to retail outlet R1001. Simultaneously the Manager of retail outlet R1001 is consolidating a report for total quantity on hand for item I1001. Sequence of the transactions are as shown below:

   What is your observation?

Time	Store Keeper's Transaction	Quantity of I1001	Manager's Transaction
10:22	Read quantity available of item 'I1001' in ware house	100	Total = 0
10:23	Transfer 25 units of I1001 to retail outlet R1001 qtyonhand = 100-25		Read qtyonhand for I1001 in ware house(100)
10:24	Write new qtyavailable	75	
10:25			Total=Total+qtyonhand(100)
10:26	Read quantity available of item 'I1001' in retail stock	20	
10:27	qtyavailable = 20+25		
10:28	Write new qtyavailable	45	
10:29	Commit		
10:30			Read quantity available of item 'I1001' in retailstock
10:31			**Total = 45 + qtyonhand(100)**
10:32		**145**	**Write Total**
10:33			**Commit**

Incorrect summary

How many units of I1001 should be present in total finally? How can this be resolved?

Estimated Time: 15 minutes

Assignment 2: Locks – Guided activity

Objective: To learn about the exclusive lock.

Background: The different locking mechanisms have been explained to you in the class. For practice, we will use the following emp table:

Table: emp with sample data

empid	empname	designation	sal
10001	George	Manager	13000
10002	Martha	Supervisor	7000
10003	Victor	Accountant	8000
10004	Edgar	Steno	5000
10005	Justin	Accountant	7500
10006	Francis	Manager	15000
10007	Albert	Manager	14500

Note: In Oracle the default locking is row exclusive lock for UPDATE/DELETE/INSERT

Step 1: Open two instances of SQL PLUS on your machine and login with your Oracle ID in both the instances.

Step 2: Go to first instance.
Write the following query:

```
UPDATE emp SET sal=9000 WHERE empid=10004;
```

Step 3: Go to the second instance.
Write the following query:

```
UPDATE emp SET sal=6000 WHERE empid=10004;
```

Note: The second instance is in wait state. The reason is the first transaction has issued the update statement for the empid 10004 and it acquires the exclusive(X) lock. The second transaction is also trying to update the same but because of the X lock acquired by the first transaction it has to wait.

Step 4: Go to the first instance and issue `COMMIT/ROLLBACK.`

Step 5: Go to the second instance. You get the SQL prompt.

Note: As soon as you `COMMIT/ROLLBACK` in the first instance the transaction is over and the X lock is released.

Step 6: Issue `COMMIT/ROLLBACK in` the second instance as well.

Summary: You have just learnt `UPDATE` DML statement acquire a row exclusive lock
The X lock is released only at the end of transaction which is marked by `COMMIT/ROLLBACK.`

Estimated time: 15 minutes

Summary of this assignment: In this assignment you have learnt the concept and usage of database locks.

Assignments on PL/SQL

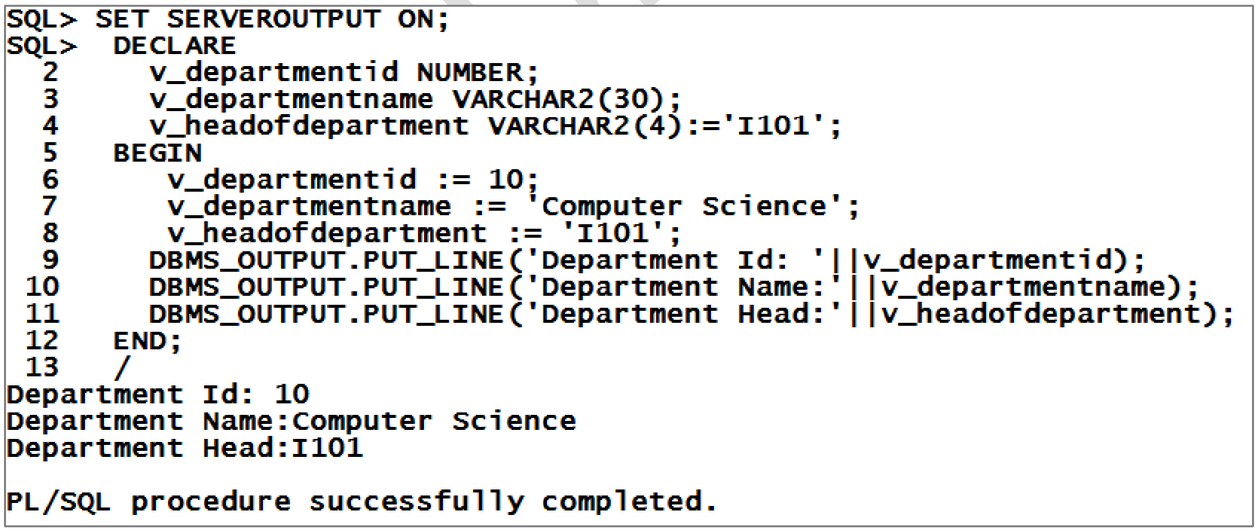
Assignment 1: PL/SQL block – Guided activity

Objective: Given a relational database, a use case representing the business requirement, be able to implement anonymous PL/SQL blocks

Background: Create the tables and insert the data using the "***CourseRegistrationDBDesign.txt***" script. If you have all the tables already created, still you can drop all the tables and recreate all the tables and populate the records so as to ensure consistency of the database.

Problem Description: Course Registration application has a department table with departmentid, departmentname and headofdepartment.

Write a PL/SQL block as shown below, which declares variables for assigning departmentid, departmentname and headofdepartment details and display the same in execution section.



```
SQL> SET SERVEROUTPUT ON;
SQL>  DECLARE
  2     v_departmentid NUMBER;
  3     v_departmentname VARCHAR2(30);
  4     v_headofdepartment VARCHAR2(4):='I101';
  5   BEGIN
  6      v_departmentid := 10;
  7      v_departmentname := 'Computer Science';
  8      v_headofdepartment := 'I101';
  9     DBMS_OUTPUT.PUT_LINE('Department Id: '||v_departmentid);
 10     DBMS_OUTPUT.PUT_LINE('Department Name:'||v_departmentname);
 11     DBMS_OUTPUT.PUT_LINE('Department Head:'||v_headofdepartment);
 12   END;
 13   /
Department Id: 10
Department Name:Computer Science
Department Head:I101

PL/SQL procedure successfully completed.
```

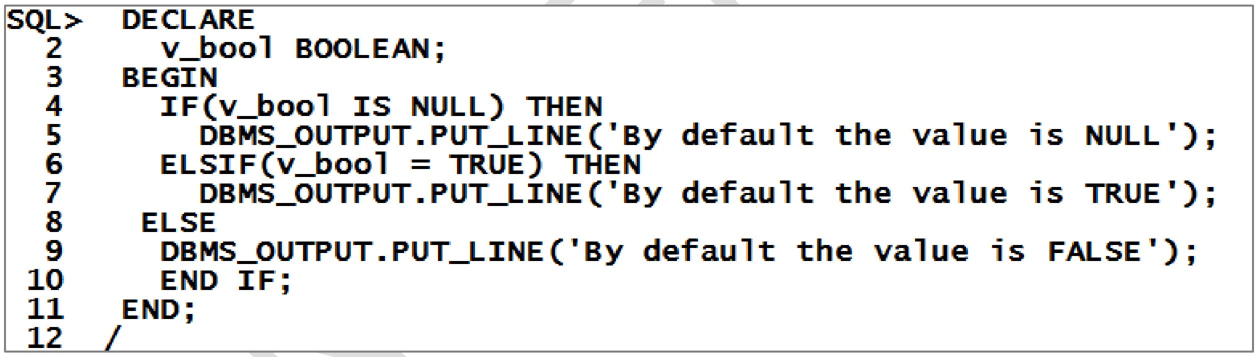
Estimated time: 15 minutes

Summary of this assignment: You have just learnt how to declare variables in the declaration section and how variables can be used in DBMS_OUTPUT.PUT_LINE statements.

Assignment 2: Usage of BOOLEAN data type – Guided activity

Objective: Given a relational database, a use case representing the business requirement, be able to implement anonymous PL/SQL blocks and understand the usage of BOOLEAN data type

Problem Description: Analyze the following code and answer the questions given below.



```
SQL>  DECLARE
  2     v_bool BOOLEAN;
  3   BEGIN
  4     IF(v_bool IS NULL) THEN
  5       DBMS_OUTPUT.PUT_LINE('By default the value is NULL');
  6     ELSIF(v_bool = TRUE) THEN
  7       DBMS_OUTPUT.PUT_LINE('By default the value is TRUE');
  8    ELSE
  9     DBMS_OUTPUT.PUT_LINE('By default the value is FALSE');
 10     END IF;
 11   END;
 12  /
```

1. What is the value of a BOOLEAN variable by default?
2. What happens if we try to print the BOOLEAN variable?
3. What are the values that can be assigned to a BOOLEAN variable?
4. What are the values expected from a BOOLEAN variable comparison?

Estimated time: 15 minutes

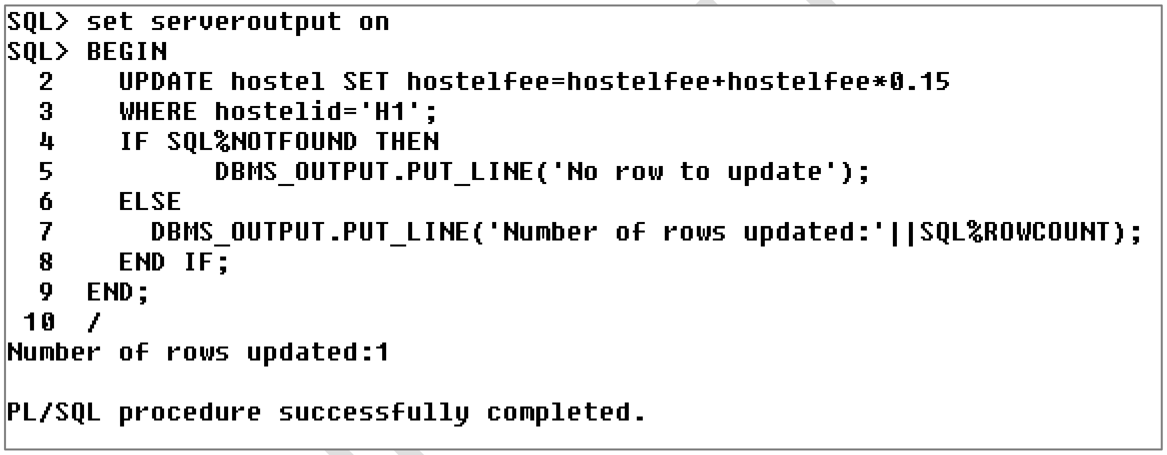
Summary of this assignment: You have just learnt the usage of Boolean data type variables.

Assignment 3: Usage of SQL% Attributes in PL/SQL – Guided activity

Objective: Given a relational database, a use case representing the business requirement, be able to implement anonymous PL/SQL blocks and use SQL% attributes in PL/SQL blocks.

Problem Description:

1. Write a PL/SQL program to update the hostelfee by 15% where hostelid is 'H1' and verify whether any row is updated and if it has affected display how many rows are updated.



```
SQL> set serveroutput on
SQL> BEGIN
      UPDATE hostel SET hostelfee=hostelfee+hostelfee*0.15
      WHERE hostelid='H1';
      IF SQL%NOTFOUND THEN
            DBMS_OUTPUT.PUT_LINE('No row to update');
      ELSE
        DBMS_OUTPUT.PUT_LINE('Number of rows updated:'||SQL%ROWCOUNT);
      END IF;
    END;
    /
Number of rows updated:1

PL/SQL procedure successfully completed.
```

2. Modify the codes of problem description 2 and 3 of previous assignment 5 and verify how many rows are affected and if any row gets affected, display the number of rows affected.

Estimated time: 30 minutes

Summary of this assignment: You have just learnt the usage of implicit cursor attributes in PLSQL blocks.