Data Warehousing ETL

TUTORIAL

1. **What does the term Heterogeneous Systems mean?**
   The term Heterogeneous System refers to system that are composed of different type of hardware, software, operating systems, and data model. Therefore, it can include a wide range of technologies with many components communicating with each other using different protocols or data format.

   Heterogeneous database system, therefore, refers to a database system that combines different types of databases, data models, and hardware platforms into a single system.

2. **What impact does data from Heterogeneous Systems have on Business Analysts?**
   Business analysts can have access to a more comprehensive and accurate analysis of business operation and trends, by integrating data from different databases and systems, they can gain a better understanding of the organization's operation.

   However, great benefits also come with great challenges, as there are many challenges for business analyst for example:
   + Data Integration: Business analysts may need to extract data from multiple databases and transform then into a common format, then load it into a single data warehouse for analysis and reporting.
   + Different data sources and regions may also make it difficult for business analysts to create an accurate report based on the current data.

3. **Business Analysts are sometimes required to run queries that access large amounts in normalised OLTP systems.**
   - **Describe possible performance issues for business analysts**
     + Slow query response time: as queries are run against large amount of data it will take times to process such demand.
     + Reduced system performance: running large queries can also affect the overall performance of the OLTP system.
     + Resource contention: as large queries can place heavy load on system resources.
   - **Describe possible performance issues for operational OLTP users**
     + Slow transaction processing and reduced system availability: as queries are run against large amount of data the users will also be affected.
     + Deadlocks and lock contention: when modifying or accessing OLTP system, multiple users may cause locks to be placed on the same data and thus cause deadlock or lock contention.
   - **Describe the reasons for the performance issues**
     + Large data volume: as OLTP are designed to handle transactional data not analytical queries.
     + Complex Query Structure: Business analyst may write complex queries that require extensive data processing.

4. **What is ETL?**
   + ETL stands for Extract, Transform, Load which is a process used to integrate data from multiple sources into a single data warehouse

5. **Why is ETL necessary?**
   ETL is necessary for:
   + Data Integration: which is used to integrate data from multiple sources.
   + Data Quality: which is to ensure that data is accurate, complete, and consistent.
   + Data Analysis: which is to ensure that data is in the same common format for data analysis.

6. **What is the ErrorEvent table and what is stored within it?**
   It is used to store information about any errors or issues that occur during the ETL process. Typically, ErrorEvent includes information such as the date and time of the error, the type of error that occurred.

7. **What is a dimension table? What are typical examples of a dimension table?**
   Categorically consistent view of data. Some examples of dimension table are: Customer, Product, Time, Location.

8. **What is a fact table?**
   A fact table is a table in relational database that contains quantitative data or facts about a specific event or transaction.

9. **What is the purpose of the DWDATE dimension table?**
   It is a special type of dimension table used in data warehousing to provide information about dates and times.

10. **What is a star schema and why is it called a star schema?**
    It is a type of database schema used in data warehousing, where data is organized around a central fact table that is linked to multiple dimension tables.

11. **Consider this normalized schema from an OLTP database.**

    SERVICE(ServId, Description)
    PK ItemId

    REGION (RegCode, RegName)
    PK RegCode

    CUST(CustId, Name, RegCode)
    PK CustId
    FK RegCode references REGION

    SALE(SaleId, ServId, CustId, Qty, ServDate, ServFee)
    PK SaleId
    FK ServId references SERVICE
    FK CustId references CUST

Suppose that a data warehouse is to be created with data provided by the Sales Database above

a) How would the customer be denormalized? (i.e. what columns would be in the cust table?)
   The CUSTOMER table would include:
   + CustId (pri)
   + Name
   + RegCode
   + RegName

b) Why would the customer data be denormalized?
To improve query performance and reduce complexity.

12. Data Warehouses typically have a table such as DW_DATE
    a. Is DW_DATE as Dimension Table or a Fact Table?
       It is a dimension table
    b. Why is such a table desirable
       It provides a standard way to track dates and time periods across data warehouse/
    c. List some of the columns that would be found in the DW_DATE table
       Date, Year, Month, Week, Quarter,….
    d. What column would be the identifier of the DW_DATE table

       It would be Date or DateId

LAB Work

1. ROWID

- Every row in every table created in Oracle has a **unique** ROWID value.
- It acts like a hidden column in the table where every row is assigned a unique value ○ it's not actually a column in the table at all
  - ROWID can be thought of as a pointer to the physical location (on disk) of the table row.
    http://download.oracle.com/docs/cd/B28359_01/server.111/b28286/pseudocolumns008.htm#sthref836
- The combination of TableName + RowID will guarantee a unique value for every value in the entire database.
- The ROWID is usually looks similar to this: **AABtmCAAEAAAURgAAC**
- Accessing a row by using its ROWID is very fast (because ROWID is a pointer to the physical location of the row)

You can display the ROWID value of some of your existing tables.

Try these SQL statements:

a. SELECT rowid, empid, empname FROM employee.

(Use the code below if the table does not exist in your database

```
22  DROP TABLE employee;
23  CREATE  TABLE employee (
24  empid INTEGER PRIMARY KEY,
25  empname VARCHAR2(50),
26  gender VARCHAR2(1),
27  salary NUMBER(6) );
28
29  INSERT INTO employee VALUES (2,'Maggie Walsh','F',60000);
30  INSERT INTO employee VALUES (5,'Wesley Wyndam-Pryce','M',75000);
31  INSERT INTO employee VALUES (7,'Harmony Kendall','F',56000);
32  INSERT INTO employee VALUES (13,'Jonathan Levinson','M',42000);
33  INSERT INTO employee VALUES (27,'Jenny Calendar','F',61000);
34
35  SET SERVEROUTPUT ON;
36  SELECT ROWID, EMPID, EMPNAME, GENDER, SALARY FROM EMPLOYEE;
37
```

Script Output | Query Result

All Rows Fetched: 5 in 0.211 seconds

	ROWID	EMPID	EMPNAME	GENDER	SALARY
1	AAEC+AAAKAAAX9kAAA	2	Maggie Walsh	F	60000
2	AAEC+AAAKAAAX9kAAB	5	Wesley Wyndam-Pryce	M	75000
3	AAEC+AAAKAAAX9kAAC	7	Harmony Kendall	F	56000
4	AAEC+AAAKAAAX9kAAD	13	Jonathan Levinson	M	42000
5	AAEC+AAAKAAAX9kAAE	27	Jenny Calendar	F	61000

```
CREATE TABLE employee (
  empid INTEGER PRIMARY KEY,
  empname VARCHAR2(50),
  gender VARCHAR2(1),
  salary NUMBER(6)     );
INSERT INTO employee VALUES (2,'Maggie Walsh','F',60000);
INSERT INTO employee VALUES (5,'Wesley Wyndam-Pryce','M',75000);
INSERT INTO employee VALUES (7,'Harmony Kendall','F',56000);
INSERT INTO employee VALUES (13,'Jonathan Levinson','M',42000);
INSERT INTO employee VALUES (27,'Jenny Calendar','F',61000);
```

2. **DML. Execute the following DDL**
   Assume that we have a business with branches in Melbourne and Sydney.
   Each business has its own staff table. They are named StaffMel and StaffSyd
   Crete these tables.

	SID	FNAME	SNAME	GENDER	SALARY	STATUS	BIRTHDAY
1	1	Jo	Dunn	F	79000	OK	26-JAN-12
2	3	Jeff	Smith	m	45000	o.k.	26-JAN-12
3	5	Sue	Jones	f	79000	OK	29-JAN-12
4	7	Dan	Brown	M	610000	Penning	02-FEB-12

	SID	FNAME	SNAME	GENDER	SALARY	STATUS	BIRTHDAY
1	2	Ben	Black	male	5417	O K	02-FEB-12
2	5	Emma	Loh	female	5920	OK	02-FEB-12
3	8	Patel	Leena	FEMALE	3580	Ok	04-FEB-12
4	9	Kelly	Down	(null)	19840	Okay	09-FEB-12

```
--drop all tables. Causes an error message if the table doesn't
exist DROP TABLE STAFFSYD CASCADE CONSTRAINTS; DROP TABLE STAFFMEL
CASCADE CONSTRAINTS;
--create the staffsyd table. It stores details of all the Sydney staff
CREATE TABLE STAFFSYD (
  SID INTEGER PRIMARY KEY,
  FNAME VARCHAR2(20),
  SNAME VARCHAR2(20),
  GENDER VARCHAR2(10),
  SALARY NUMBER,
  STATUS VARCHAR2(10),

  BIRTHDATE DATE );
--create the staffmel table. It stores details of all the Melbourne staff
CREATE TABLE STAFFMEL (
  SID INTEGER PRIMARY KEY,
  FNAME VARCHAR2(20),
  SNAME VARCHAR2(20),
  GENDER VARCHAR2(10),
  SALARY NUMBER,
  STATUS VARCHAR2(10),
  BIRTHDATE DATE );
```

3. DML. Execute the following DML

Create some data for each table.

--Insert all the Sydney staff

```
INSERT INTO STAFFMEL VALUES(1,'Jo','Dunn','F',79000,'OK','26-JAN-2012'); INSERT
INTO STAFFMEL VALUES(3,'Jeff','Smith','m',45000,'o.k.','26-JAN-2012'); INSERT
INTO STAFFMEL VALUES(5,'Sue','Jones','f',79000,'OK','29-JAN-2012'); INSERT INTO
STAFFMEL VALUES(7,'Dan','Brown','M',610000,'Penning','02-FEB-2012');
```

--Insert all the Melbourne staff

```
INSERT INTO STAFFSYD VALUES(2,'Ben','Black','male',5417,'O K', '02-FEB-2012');
INSERT INTO STAFFSYD VALUES(5,'Emma','Loh','female',5920,'OK', '02-FEB-2012');
INSERT INTO STAFFSYD VALUES(8,'Patel','Leena','FEMALE',3580,'Ok', '04-FEB-2012');
INSERT INTO STAFFSYD VALUES(9,'Kelly','Down',null,19840,'Okay', '09-FEB-2012');
```

Our aim is to create a create a Query that lists all staff details PLUS the ROWID and the TABLENAME of every row of these two tables

```
SOURCE_ROWID       SOURCE_TABLE SID FNAME   SNAME GENDER SALARY  STATUS
------------------ ---------- ------------------------------------
AABtnnAAEAAAURgAAA STAFFMEL     1   Jo      Dunn  F      79000   OK
AABtnnAAEAAAURgAAB STAFFMEL     3   Jeff    Smith m      45000   o.k.
AABtnnAAEAAAURgAAC STAFFMEL     5   Sue     Jones f      79000   OK
AABtnnAAEAAAURgAAD STAFFMEL     7   Dan     Brown M      610000  Penning
AABtnlAAEAAAT1wAAA STAFFSYD     2   Ben     Black male   5417    O K
AABtnlAAEAAAT1wAAB STAFFSYD     5   Emma    Loh   female 5920    OK
AABtnlAAEAAAT1wAAC STAFFSYD     8   Patel   Leena FEMALE 3580    Ok

AABtnlAAEAAAT1wAAD STAFFSYD     9   Kelly   Down  null   19840   Okay
```

Follow these steps:

4. List data from staffmel table

Write a query to list all rows in the staffmel table.

- Column 1 must display the **ROWID** of the row
- Column 2 must display the literal text **'STAFFMEL'**
- Column 3-8 must display SID, FNAME, SNAME, GENDER, SALARY, STATUS, BRITHDAY

Script Output | Query Result

All Rows Fetched: 4 in 0.252 seconds

	ROWID	TABLE_NAME	SID	FNAME	SNAME	GENDER	SALARY	STATUS	BIRTHDAY
1	AAEB5sAAKAAAY3fAAA	STAFFMEL	1	Jo	Dunn	F	79000	OK	26-JAN-12
2	AAEB5sAAKAAAY3fAAB	STAFFMEL	3	Jeff	Smith	m	45000	o.k.	26-JAN-12
3	AAEB5sAAKAAAY3fAAC	STAFFMEL	5	Sue	Jones	f	79000	OK	29-JAN-12
4	AAEB5sAAKAAAY3fAAD	STAFFMEL	7	Dan	Brown	M	610000	Penning	02-FEB-12

```
ROWID              TABLE_NAME SID FNAME   SNAME GENDER SALARY STATUS   BIRTHDAY
                   ---------- ----------------------------------------------
AABtnnAAEAAAURgAAA STAFFMEL    1   Jo      Dunn  F      79000  OK       26-JAN-2012
AABtnnAAEAAAURgAAB STAFFMEL    3   Jeff    Smith m      45000  o.k.     26-JAN-2012
AABtnnAAEAAAURgAAC STAFFMEL    5   Sue     Jones f      79000  OK       29-JAN-2012

AABtnnAAEAAAURgAAD STAFFMEL    7   Dan     Brown M      610000 Penning  02-FEB-2012
```

5. List data from staffsyd table

Write a query to list all rows in the staffsyd table in the same format as above.

Query Result

All Rows Fetched: 4 in 0.248 seconds

	ROWID	SOURCE_TABLE	SID	FNAME	SNAME	GENDER	SALARY	STATUS	BIRTHDAY
1	AAEB5qAAKAAAY3/AAA	STAFFSYD	2	Ben	Black	male	5417	O K	02-FEB-12
2	AAEB5qAAKAAAY3/AAB	STAFFSYD	5	Emma	Loh	female	5920	OK	02-FEB-12
3	AAEB5qAAKAAAY3/AAC	STAFFSYD	8	Patel	Leena	FEMALE	3580	Ok	04-FEB-12
4	AAEB5qAAKAAAY3/AAD	STAFFSYD	9	Kelly	Down	(null)	19840	Okay	09-FEB-12

6. **Create a VIEW named staff_all_view to list all the rows from both tables If**

   you think you can do this without help, jump to Step 8 below.

Query Result

All Rows Fetched: 8 in 0.244 seconds

	SOURCE_ROWID	TABLE_NAME	SID	FNAME	SNAME	GENDER	SALARY	STATUS	BIRTHDAY
1	AAEB5qAAKAAAY3/AAA	STAFFSYD	2	Ben	Black	male	5417	O K	02-FEB-12
2	AAEB5qAAKAAAY3/AAB	STAFFSYD	5	Emma	Loh	female	5920	OK	02-FEB-12
3	AAEB5qAAKAAAY3/AAC	STAFFSYD	8	Patel	Leena	FEMALE	3580	Ok	04-FEB-12
4	AAEB5qAAKAAAY3/AAD	STAFFSYD	9	Kelly	Down	(null)	19840	Okay	09-FEB-12
5	AAEB5sAAKAAAY3fAAA	STAFFMEL	1	Jo	Dunn	F	79000	OK	26-JAN-12
6	AAEB5sAAKAAAY3fAAB	STAFFMEL	3	Jeff	Smith	m	45000	o.k.	26-JAN-12
7	AAEB5sAAKAAAY3fAAC	STAFFMEL	5	Sue	Jones	f	79000	OK	29-JAN-12
8	AAEB5sAAKAAAY3fAAD	STAFFMEL	7	Dan	Brown	M	610000	Penning	02-FEB-12

7. Use a SELECT UNION to list rows from both the staffmel and staffsyd tables

   a. Use the select statement from step 5 above:

```
SELECT ROWID,'STAFFMEL' AS "SOURCE_TABLE",
SID,FNAME,SNAME,GENDER,SALARY,STATUS,BRITHDAY FROM STAFFMEL
```

Query Result

All Rows Fetched: 4 in 0.245 seconds

	ROWID	SOURCE_TABLE	SID	FNAME	SNAME	GENDER	SALARY	STATUS	
1	AAEB5sAAKAAAY3fAAA	STAFFMEL	1	Jo	Dunn	F	79000	OK	26
2	AAEB5sAAKAAAY3fAAB	STAFFMEL	3	Jeff	Smith	m	45000	o.k.	26
3	AAEB5sAAKAAAY3fAAC	STAFFMEL	5	Sue	Jones	f	79000	OK	29
4	AAEB5sAAKAAAY3fAAD	STAFFMEL	7	Dan	Brown	M	610000	Penning	02

   b. Copy, Paste and Modify the select statement from step 7a above so that it uses the staffsyd table

```
SELECT ROWID,'STAFFSYD' AS "SOURCE_TABLE",
SID,FNAME,SNAME,GENDER,SALARY,STATUS,BRITHDAY FROM STAFFSYD
```

	ROWID	SOURCE_TABLE	SID	FNAME	SNAME	GENDER	SALARY	STATUS	BIRTH
1	AAEB5qAAKAAAY3/AAA	STAFFSYD	2	Ben	Black	male	5417	O K	02-FEB
2	AAEB5qAAKAAAY3/AAB	STAFFSYD	5	Emma	Loh	female	5920	OK	02-FEB
3	AAEB5qAAKAAAY3/AAC	STAFFSYD	8	Patel	Leena	FEMALE	3580	Ok	04-FEB
4	AAEB5qAAKAAAY3/AAD	STAFFSYD	9	Kelly	Down	(null)	19840	Okay	09-FEB

   c. Use the statements from 7a and 7b and separate them with a UNION clause
   Note: Make sure that you remove any semi colon before the UNION clause.

```
SELECT ROWID,'STAFFMEL' AS "SOURCE_TABLE",
SID,FNAME,SNAME,GENDER,SALARY,STATUS,BRITHDAY FROM STAFFMEL
UNION
SELECT ROWID,'STAFFSYD' AS "SOURCE_TABLE",
SID,FNAME,SNAME,GENDER,SALARY,STATUS,BRITHDAY FROM STAFFSYD
```

Query Result

All Rows Fetched: 8 in 0.246 seconds

	ROWID	SOURCE_TABLE	SID	FNAME	SNAME	GENDER	SALARY	STATUS	BIRTH
1	AAEB5qAAKAAAY3/AAA	STAFFSYD	2	Ben	Black	male	5417	O K	02-FEB
2	AAEB5qAAKAAAY3/AAB	STAFFSYD	5	Emma	Loh	female	5920	OK	02-FEB
3	AAEB5qAAKAAAY3/AAC	STAFFSYD	8	Patel	Leena	FEMALE	3580	Ok	04-FEB
4	AAEB5qAAKAAAY3/AAD	STAFFSYD	9	Kelly	Down	(null)	19840	Okay	09-FEB
5	AAEB5sAAKAAAY3fAAA	STAFFMEL	1	Jo	Dunn	F	79000	OK	26-JAN
6	AAEB5sAAKAAAY3fAAB	STAFFMEL	3	Jeff	Smith	m	45000	o.k.	26-JAN
7	AAEB5sAAKAAAY3fAAC	STAFFMEL	5	Sue	Jones	f	79000	OK	29-JAN
8	AAEB5sAAKAAAY3fAAD	STAFFMEL	7	Dan	Brown	M	610000	Penning	02-FEB

d. Now create a view based on the select…union statement

```
Create or replace view STAFF_ALL_VIEW
     (SOURCE_ROWID, TABLE_NAME, SID, FNAME, SNAME,
      GENDER, SALARY, STATUS,BRITHDAY ) AS SELECT
ROWID,'STAFFMEL' AS "SOURCE_TABLE",
SID,FNAME,SNAME,GENDER,SALARY,STATUS,BRITHDAY FROM STAFFMEL
UNION
SELECT ROWID,'STAFFSYD' AS "SOURCE_TABLE",
SID,FNAME,SNAME,GENDER,SALARY,STATUS,BRITHDAY FROM STAFFSYD
```

SQL | All Rows Fetched: 8 in 0.257 seconds

	SOURCE_ROWID	TABLE_NAME	SID	FNAME	SNAME	GENDER	SALARY	STATUS	BIRTHDA
1	AAEB5qAAKAAAY3/AAA	STAFFSYD	2	Ben	Black	male	5417	O K	02-FEB-1
2	AAEB5qAAKAAAY3/AAB	STAFFSYD	5	Emma	Loh	female	5920	OK	02-FEB-1
3	AAEB5qAAKAAAY3/AAC	STAFFSYD	8	Patel	Leena	FEMALE	3580	Ok	04-FEB-1
4	AAEB5qAAKAAAY3/AAD	STAFFSYD	9	Kelly	Down	(null)	19840	Okay	09-FEB-1
5	AAEB5sAAKAAAY3fAAA	STAFFMEL	1	Jo	Dunn	F	79000	OK	26-JAN-1
6	AAEB5sAAKAAAY3fAAB	STAFFMEL	3	Jeff	Smith	m	45000	o.k.	26-JAN-1
7	AAEB5sAAKAAAY3fAAC	STAFFMEL	5	Sue	Jones	f	79000	OK	29-JAN-1
8	AAEB5sAAKAAAY3fAAD	STAFFMEL	7	Dan	Brown	M	610000	Penning	02-FEB-1

e. Now select all the rows from STAFF_ALL_VIEW

```
SOURCE_ROWID       SOURCE_TABLE SID FNAME   SNAME GENDER SALARY   STATUS
------------------  ---------- ------------------------------------
AABtnnAAEAAAURgAAA STAFFMEL     1   Jo      Dunn  F      79000    OK
AABtnnAAEAAAURgAAB STAFFMEL     3   Jeff    Smith m      45000    o.k.
AABtnnAAEAAAURgAAC STAFFMEL     5   Sue     Jones f      79000    OK
AABtnnAAEAAAURgAAD STAFFMEL     7   Dan     Brown M      610000   Penning
AABtnlAAEAAAT1wAAA STAFFSYD     2   Ben     Black male   5417     O K
AABtnlAAEAAAT1wAAB STAFFSYD     5   Emma    Loh   female 5920     OK
AABtnlAAEAAAT1wAAC STAFFSYD     8   Patel   Leena FEMALE 3580     Ok

AABtnlAAEAAAT1wAAD STAFFSYD     9   Kelly   Down  null   19840    Okay
```

Next step:
Write code that checks for invalid data in the sources tables.
If invalid data is found, store details about the offending rows into the ERROR_EVENT table

8. Create the Error Event table.

```
DROP TABLE ERROR_EVENT CASCADE CONSTRAINTS;
--create the error_event table
CREATE TABLE ERROR_EVENT (
  SOURCE_TABLE VARCHAR2(20),
  SOURCE_ROWID  ROWID,
  FILTER_ID NUMBER(2),
  DATE_TIME DATE,
  ACTION VARCHAR2(6),
  CHECK (ACTION IN ('SKIP', 'MODIFY')),
  CONSTRAINT ERROR_EVENT_PK PRIMARY KEY (SOURCE_TABLE, SOURCE_ROWID, FILTER_ID));
```

9. Filter 1.

- Create Filter 1 that determines if a staff row has a null gender.

10. Determine which rows have a null gender

a. Write a query based on staff_all_view

```
SELECT SID, GENDER
FROM STAFF_ALL_VIEW
WHERE GENDER IS NULL;
```

b. Modify the above query.

- Replace the columns in the select clause with ROWID and SOURCE_TABLE

```
SELECT SOURCE_ROWID, SOURCE_TABLE
FROM STAFF_ALL_VIEW
WHERE GENDER IS NULL;
```

11. Insert invalid row details into the Error_Event table

a. Modify the query above so that it adds the **filter number**, **sysdate** and **action** to the result set:

```
SOURCE_ROWID        SOURCE_TABLE FILTERID SYSDATE    'SKIP'
                    ---------- ------- ---------- ------
AABtnlAAEAAAT1wAAB----------------- STAFFSYD     1        18/APR/12  SKIP
AABtnnAAEAAAURgAAC STAFFMEL     1        18/APR/12  SKIP
```

b. Need help writing the Select statement?

```
SELECT SOURCE_ROWID, SOURCE_TABLE, 1 AS "FILTERID",
       SYSDATE,  'SKIP' AS ACTION
FROM STAFF_ALL
WHERE GENDER IS NULL;
```

c. Convert the SELECT statement above into an INSERT INTO statement so that adds rows to the error_event table.

```
INSERT INTO ERROR_EVENT
(SOURCE_ROWID, TABLE_NAME, FILTER_ID, DATE_TIME, ACTION)
SELECT SOURCE_ROWID, SOURCE_TABLE, 1 AS "FILTER",
       SYSDATE,  'SKIP' AS ACTION
FROM STAFF_ALL
WHERE GENDER IS NULL;
```

12. List the contents of the Error_Event table

a) Write a query to display the contents of the error_event table.
Your output should look similar to this (rowids will be different)

```
SOURCE_TABLE SOURCE_ROWID        FILTER_ID  DATE_TIME    ACTION
             ------------------ ---------- ------------ ------

STAFFSYD---------- AABtmAAAEAAAT1wAAB 1          18/APR/12    SKIP
```

b) Modify the above query so that only the TABLE_NAME and SOURCE_ROWID are displayed (this query will be useful later)

```
SOURCE_TABLE SOURCE_ROWID
------------ ------------
STAFFSYD     AABtmAAAEAAAT1wAAB
```

13. Dealing with Upper and Lower case combinations

Copy the code that creates the STAFF_ALL view.
Call the new view STAFF_ALL_UPPER_VIEW

Modify the code and use the UPPER() function so that so that the STATUS value is always displayed in uppercase.

```
Create or replace view STAFF_ALL_UPPER_VIEW
     (SOURCE_ROWID, TABLE_NAME, SID, FNAME, SNAME,
      GENDER, SALARY, STATUS,BRITHDAY ) AS SELECT
ROWID,'STAFFMEL' AS "SOURCE_TABLE",
SID,UPPER(FNAME),UPPER(SNAME),GENDER,SALARY,STATUS,BRITHDAY FROM
STAFFMEL
UNION
SELECT ROWID,'STAFFSYD' AS "SOURCE_TABLE", SID,

UPPER(FNAME),UPPER(SNAME),GENDER,SALARY,STATUS,BRITHDAY FROM STAFFSYD
```

```
SELECT * FROM STAFF_ALL_UPPER_VIEW;
SOURCE_ROWID        TABLE_NAME SID FNAME    SNAME  GENDER SALARY   STATUS
--------------------- ----------------------------- ----------------
AAAp1GAAEAAACXMAAA STAFFSYD   2   Ben     Black  male   5417     O K
AAAp1GAAEAAACXMAAB STAFFSYD   5   Emma    Loh    female 5920     OK
AAAp1GAAEAAACXMAAC STAFFSYD   8   Patel   Leena  FEMALE 3580     OK
AAAp1GAAEAAACXMAAD STAFFSYD   9   Kelly   Down   NULL   19840    OKAY
AAAp1IAAEAAACWkAAA STAFFMEL   1   Jo      Dunn   F      79000    OK
AAAp1IAAEAAACWkAAB STAFFMEL   3   Jeff    Smith  m      45000    O.K.
AAAp1IAAEAAACWkAAC STAFFMEL   5   Sue     Jones  f      79000    OK
AAAp1IAAEAAACWkAAD STAFFMEL   7   Dan     Brown  M      610000   PENNING

 8 rows selected
```

```
--PART 13--

Create or replace view STAFF_ALL_UPPER_VIEW
(SOURCE_ROWID, TABLE_NAME, SID, FNAME, SNAME,
GENDER, SALARY, STATUS ,BIRTHDAY ) AS
SELECT ROWID,'STAFFMEL' AS "SOURCE_TABLE",
SID,UPPER(FNAME),UPPER(SNAME),GENDER,SALARY,STATUS ,BIRTHDAY FROM
STAFFMEL
UNION
SELECT ROWID,'STAFFSYD' AS "SOURCE_TABLE", SID,
UPPER(FNAME),UPPER(SNAME),GENDER,SALARY,STATUS ,BIRTHDAY FROM STAFFSYD;

SELECT * FROM STAFF_ALL_UPPER_VIEW;

--PART 14--
```

Script Output | Query Result

All Rows Fetched: 8 in 0.211 seconds

	SOURCE_ROWID	TABLE_NAME	SID	FNAME	SNAME	GENDER	SALARY	STATUS	BIRTHDAY
1	AAEB5qAAKAAAY3/AAA	STAFFSYD	2	BEN	BLACK	male	5417	O K	02-FEB-12
2	AAEB5qAAKAAAY3/AAB	STAFFSYD	5	EMMA	LOH	female	5920	OK	02-FEB-12
3	AAEB5qAAKAAAY3/AAC	STAFFSYD	8	PATEL	LEENA	FEMALE	3580	Ok	04-FEB-12
4	AAEB5qAAKAAAY3/AAD	STAFFSYD	9	KELLY	DOWN	(null)	19840	Okay	09-FEB-12
5	AAEB5sAAKAAAY3fAAA	STAFFMEL	1	JO	DUNN	F	79000	OK	26-JAN-12
6	AAEB5sAAKAAAY3fAAB	STAFFMEL	3	JEFF	SMITH	m	45000	o.k.	26-JAN-12
7	AAEB5sAAKAAAY3fAAC	STAFFMEL	5	SUE	JONES	f	79000	OK	29-JAN-12
8	AAEB5sAAKAAAY3fAAD	STAFFMEL	7	DAN	BROWN	M	610000	Penning	02-FEB-12

14. Check Spelling

Create a table named STATUS_SPELLING that contains all of the **incorrect** spelling variations of 'OK' and 'PENDING'

a. Create a table named STATUS_SPELLING

```
CREATE TABLE STATUS_SPELLING (
  BAD_STATUS VARCHAR2(20),
  GOOD_STATUS VARCHAR2(20) );
```

b. Insert the following data to the table:

```
INSERT INTO STATUS_SPELLING VALUES ('O.K.','OK');
INSERT INTO STATUS_SPELLING VALUES ('PENNING','PENDING');
INSERT INTO STATUS_SPELLING VALUES ('O K','OK');
INSERT INTO STATUS_SPELLING VALUES ('OKAY','OK');
```

15. Fix STATUS spelling

These steps show how to determine the correct spelling of misspelt words.

a. Write a query to list the SID & STATUS columns of the STAFF_ALL_UPPER_VIEW

```
SID  STATUS
-------------
 2    O K
 9    OKAY
 3    O.K.
7PENNING
```

```
SELECT SID, STATUS
FROM STAFF_ALL_UPPER_VIEW

SELECT SID, STATUS
FROM STAFF_ALL_UPPER_VIEW
WHERE STATUS IN ( SELECT BAD_STATUS FROM STATUS_SPELLING);

--CORRECT--
SELECT SA.SID, SA.STATUS, SS.BAD_STATUS
FROM STAFF_ALL_UPPER_VIEW SA
INNER JOIN STATUS_SPELLING SS
```

Script Output | Query Result

All Rows Fetched: 8 in 0.171 seconds

	SID	STATUS
1	2	O K
2	5	OK
3	8	Ok
4	9	Okay
5	1	OK
6	3	o.k.
7	5	OK
8	7	Penning

b. Modify the query so that it lists data from the STAFF_ALL_UPPER_VIEW and STATUS_SPELLING tables

- Use aliases: SA for STAFF_ALL_UPPER and SS for STATUS_SPELLING
- Display the SA.SID, SA.STATUS and SS.BAD_STATUS columns where SA.STATUS = SS.BAD_STATUS

```
--CORRECT--
SELECT SA.SID, SA.STATUS, SS.BAD_STATUS
FROM STAFF_ALL_UPPER_VIEW SA, STATUS_SPELLING SS
WHERE SA.STATUS = SS.BAD_STATUS;

SELECT SA.SID, SA.STATUS, SS.GOOD_STATUS
FROM STAFF_ALL_UPPER_VIEW SA, STATUS_SPELLING SS
WHERE SA.STATUS = SS.GOOD_STATUS;

SELECT SA.SID, SA.STATUS, SS.GOOD_STATUS
```

Query...

All Rows Fetched: 1 in 0.168 seconds

	SID	STATUS	BAD_STATUS
1	2	O K	O K

```
SID   STATUS       BAD_STATUS
--------------- -----------
2     O K          O K
9     OKAY         OKAY
3     O.K.         O.K.

7     PENNING      PENNING
```

c. Modify the query so that it lists the GOOD_STATUS column value instead of the BAD_STATUS column

- Use aliases: SA for STAFF_ALL_UPPER_VIEW and SS for STATUS_SPELLING
- Display the SA.SID, SA.STATUS and SS.GOOD_STATUS columns where SA.STATUS = SS.BAD_STATUS

```
WHERE SA.STATUS = SS.BAD_STATUS;

SELECT SA.SID, SA.STATUS, SS.GOOD_STATUS
FROM STAFF_ALL_UPPER_VIEW SA, STATUS_SPELLING SS
WHERE SA.STATUS = SS.GOOD_STATUS;

SELECT SA.SID, SA.STATUS, SS.GOOD_STATUS



SELECT SA.SID, SA.STATUS, SS.GOOD_STATUS
FROM STAFF_ALL_UPPER_VIEW SA
NATURAL JOIN STATUS_SPELLING SS
```

Query Result

All Rows Fetched: 9 in 0.169 seconds

	SID	STATUS	GOOD_STATUS
1	5	OK	OK
2	5	OK	OK
3	5	OK	OK
4	1	OK	OK
5	1	OK	OK
6	1	OK	OK
7	5	OK	OK
8	5	OK	OK
9	5	OK	OK

```
SID   STATUS        GOOD_STATUS
--------------- ------------
2     O K           OK
9     OKAY          OK
3     O.K.          OK

7     PENNING       PENDING
```

16. Modify the script

Your script must now implement Filter 2.

All rows that have bad Status spelling must be added to the Error_Event table.

```
-- Insert bad status spelling rows into Error_Event table
INSERT INTO Error_Event(SID, STATUS)
SELECT SA.SID, SA.STATUS
FROM STAFF_ALL_UPPER_VIEW SA, STATUS_SPELLING SS
WHERE SA.STATUS = SS.BAD_STATUS;

SELECT * FROM ERROR_EVENT
```

Query Result

All Rows Fetched: 2 in 0.172 seconds

	SOURCE_TABLE	SOURCE_ROWID	FILTER_ID	DATE_TIME	ACTION
1	STAFF	AAEB7oAAKAAAY4TAAD	1	29-MAR-23	SKIP
2	STAFF	AAEB5qAAKAAAY3/AAD	1	29-MAR-23	SKIP