

**Department of Software Engineering  
Mehran University of Engineering and Technology, Jamshoro**

**Course: SWE324 - Data Warehousing and Data Mining**

|                   |                |                          |                |
|-------------------|----------------|--------------------------|----------------|
| <b>Instructor</b> | Rabeea Jaffari | <b>Practical/Lab No.</b> | 03             |
| <b>Date</b>       | 16 April 2019  | <b>CLOs</b>              | CLO-4: P3 & P4 |
| <b>Signature</b>  |                | <b>Assessment Score</b>  | 1 Mark         |

|                   |  |
|-------------------|--|
| <b>Topic</b>      | To familiar with OLTP system query execution |
| <b>Objectives</b> | - To learn query execution in OLTP databases |

**Lab Discussion: Theoretical concepts and Procedural steps**

**SQL QUERIES:**

**Solution:**

1. Write An SQL Query To Fetch "FIRST\_NAME" From Worker Table Using The Alias Name As <WORKER\_NAME>.

**Select FIRST\_NAME AS WORKER\_NAME from Worker;**

2. Write An SQL Query To Fetch "FIRST\_NAME" From Worker Table In Upper Case.

**Select UPPER(FIRST\_NAME) From Worker;**

3. Write An SQL Query To Fetch Unique Values Of DEPARTMENT From Worker Table.

**Select DISTINCT DEPARTMENT from Worker;**

4. Write An SQL Query To Print Last Three Characters Of FIRST\_NAME From Worker Table.

**Select RIGHT(FIRST\_NAME, 3) from Worker;**

5. Write An SQL Query To Print The FIRST\_NAME From Worker Table After Removing White Spaces From The Left Side.

**SELECT LTRIM(FIRST\_NAME) FROM Worker;**

6. Write An SQL Query To Print The FIRST\_NAME From Worker Table After Replacing 'a' With 'A'.

**SELECT REPLACE(' FIRST\_NAME', 'a', 'A') FROM Worker;**

7. Write An SQL Query To Print All Worker Details From The Worker Table Order By FIRST\_NAME Descending.

**SELECT \* FROM Worker ORDER BY FIRST\_NAME DESC;**

8. Write An SQL Query To Print Details For Workers With The First Name As "Vipul" And "Satish" From Worker Table.

**SELECT \* from Worker WHERE FIRST\_NAME IN('Vipul','Satish');**

9. Write An SQL Query To Print Details Of Workers With DEPARTMENT Name As "Admin".

**SELECT \*FROM Worker WHERE DEPARTMENT= "Admin";**

10. Write An SQL Query To Print Details Of The Workers Whose SALARY Lies Between 100000 And 500000.

**SELECT \* FROM Worker WHERE SALARY BETWEEN 100000 AND 500000;**

11. Write An SQL Query To Print Details Of The Workers Who Have Joined In Jan'2014.

**SELECT \* FROM Worker WHERE JOINING\_DATE LIKE '2014-01-%';**

12. Write An SQL Query To Fetch The Count Of Employees Working In The Department 'Admin'.

**SELECT COUNT(WORKER\_ID) FROM Worker WHERE DEPARTMENT='Admin';**

13. Write An SQL Query To Fetch The No. Of Workers For Each Department In The Ascending Order.

**SELECT COUNT(WORKER\_ID ), DEPARTMENT FROM Worker  
GROUP BY DEPARTMENT;**

14. Write An SQL Query To Fetch The List Of Employees With The Same Salary.

**Select distinct W.WORKER\_ID, W.FIRST\_NAME, W.Salary  
from Worker W, Worker W1  
where W.Salary = W1.Salary  
and W.WORKER\_ID != W1.WORKER\_ID;**

15. Write An SQL Query To Show The Second Highest Salary From A Table.

**Select max(Salary) from Worker  
where Salary not in (Select max(Salary) from Worker;**

## Lab Tasks

**Submission Date: 26-04-18**

Execute OLTP queries for the following scenario:

*An organization has a main data table that lists events. Each event has a series of fields including duration, reason code and status.*

*The OLTP system has a table of statuses with attributes status code and Name and there is a common field between the main table (status code), which allows the organization to print the status name rather than the code.*

*There are four statuses (ready, delay, spare and down) and there are a range of reasons associated with each status with reason codes. It is possible for the same reason code to exist in two statuses (e.g. delay maintenance and down maintenance).*

*For some reason, the source database has four tables of reasons (one for each status). Each of these has a field which links to the main reason code.*

*The organization wishes to do the following:*

*When the main table record status is 'delay', then retrieve the name in the 'delay' table which corresponds to the main table delay code. When the status is 'down', do the same thing but look up the name in the 'down' table.*

## Sample screenshots:

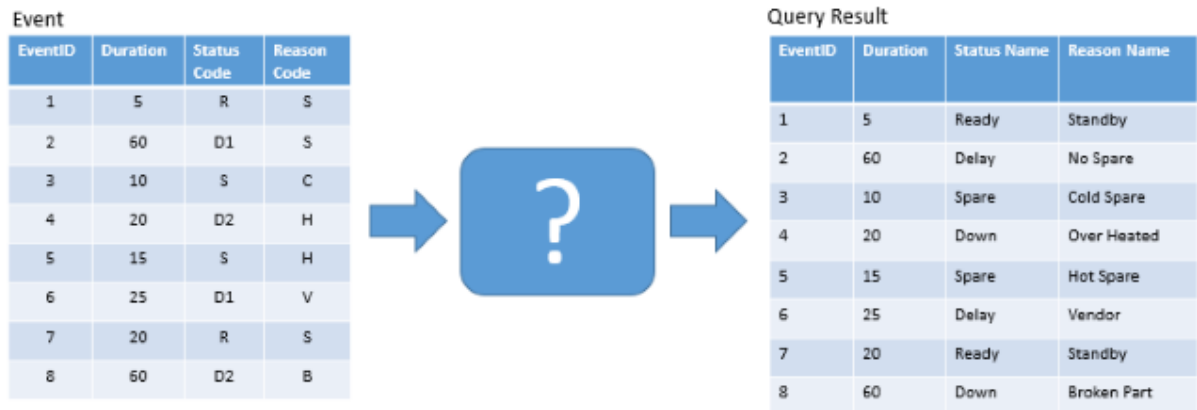
### Matching Challenges

| Event   |          |             |             | Status      |       | ReadyReason |         | DelayReason |          | SpareReason |            | DownReason  |             |
|---------|----------|-------------|-------------|-------------|-------|-------------|---------|-------------|----------|-------------|------------|-------------|-------------|
| EventID | Duration | Status Code | Reason Code | Status Code | Name  | Reason Code | Name    | Reason Code | Name     | Reason Code | Name       | Reason Code | Name        |
| 1       | 5        | R           | S           | R           | Ready | S           | Standby | R           | Running  | C           | Cold Spare | B           | Broken Part |
| 2       | 60       | D1          | S           | D1          | Delay |             |         |             |          | H           | Hot Spare  | H           | Over Heated |
| 3       | 10       | S           | C           | S           | Spare |             |         | S           | No Spare |             |            |             |             |
| 4       | 20       | D2          | H           | D2          | Down  |             |         | V           | Vendor   |             |            |             |             |
| 5       | 15       | S           | H           |             |       |             |         |             |          |             |            |             |             |
| 6       | 25       | D1          | V           |             |       |             |         |             |          |             |            |             |             |
| 7       | 20       | R           | S           |             |       |             |         |             |          |             |            |             |             |
| 8       | 60       | D2          | B           |             |       |             |         |             |          |             |            |             |             |

- Getting the Reason Name is much more difficult.
- The table to match ReasonCode depends on the StatusCode.
- There isn't a built in way to do this type of join.
- However, if we had a table of StatusCode, ReasonCode, and Names, we could match on the two columns.

Query output should be:

Goal: For each event display the EventID, Duration, Status Name, and Reason Name.



Problem:

- Each status code has a corresponding table of reason codes.
- To get the name of the reason code, you first need to use the status code to determine which reason table to match.
- These tables aren't normalized, but sometimes you get what you get and just have to cope...

Hint: Look regarding joins and union in sql.

Solution:

EVENT

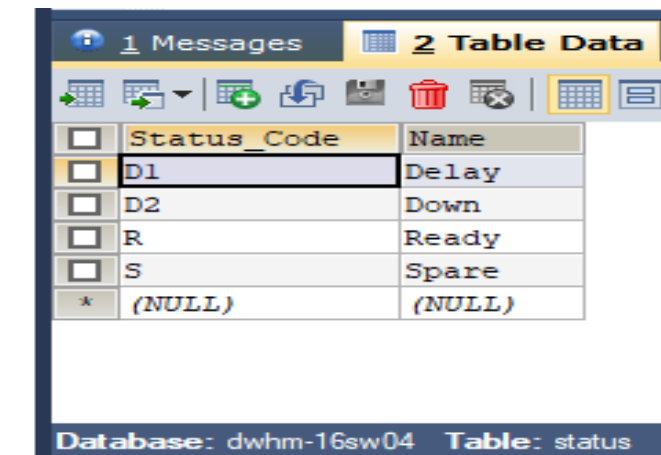
1 Messages

2 Table Data

3 Info

<

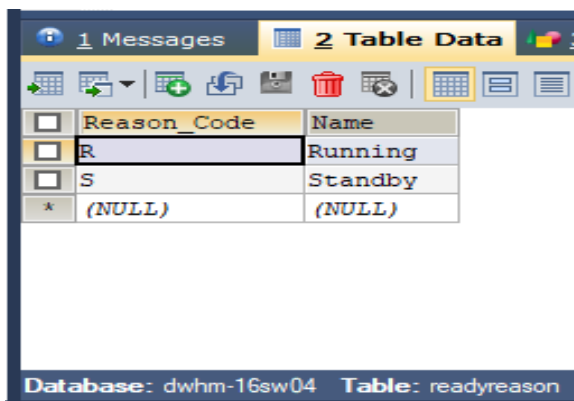
## STATUS



| Status_Code | Name   |
|-------------|--------|
| D1          | Delay  |
| D2          | Down   |
| R           | Ready  |
| S           | Spare  |
| *           | (NULL) |

Database: dwhm-16sw04 Table: status

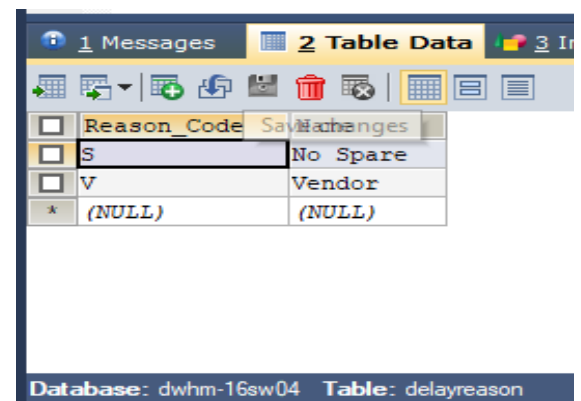
## ReadyReason



| Reason_Code | Name    |
|-------------|---------|
| R           | Running |
| S           | Standby |
| *           | (NULL)  |

Database: dwhm-16sw04 Table: readyreason

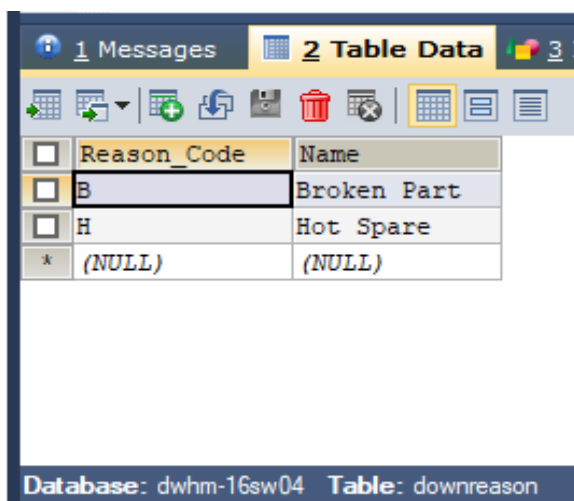
## DelayReason



| Reason_Code | Name     |
|-------------|----------|
| S           | No Spare |
| V           | Vendor   |
| *           | (NULL)   |

Database: dwhm-16sw04 Table: delayreason

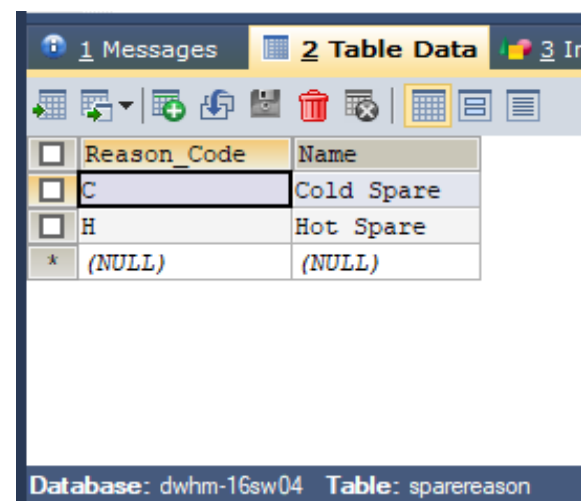
## DownReason



| Reason_Code | Name        |
|-------------|-------------|
| B           | Broken Part |
| H           | Hot Spare   |
| *           | (NULL)      |

Database: dwhm-16sw04 Table: downreason

## SpareReason



| Reason_Code | Name       |
|-------------|------------|
| C           | Cold Spare |
| H           | Hot Spare  |
| *           | (NULL)     |

Database: dwhm-16sw04 Table: sparereason

Query

History

delayreason

downreason

event

readyreason

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matching Tags. [Ctrl+Shift+S]

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

```

/*16SW04*/ SELECT EventID, Duration, S.Name AS Status_Name, SR.Name AS Reason_Name
FROM EVENT E
INNER JOIN STATUS S
ON E.Status_Code = S.Status_Code
INNER JOIN (
SELECT 'R' AS Status_Code, Reason_Code, NAME
FROM ReadyReason
UNION
SELECT 'D1', Reason_Code, NAME
FROM DelayReason
UNION
SELECT 'S', Reason_Code, NAME
FROM SpareReason
UNION
SELECT 'D2', Reason_Code, NAME
FROM DownReason
) SR
ON E.Status_Code = SR.Status_Code
AND E.Reason_Code = SR.Reason_Code
ORDER BY EVENTID; /*16SW04*//*Jawaria Dhakhan*/

```

1 Result

2 Profiler

3 Messages

4 Table Data

5 Info

(Read Only)

|                          | EventID | Duration | Status_Name | Reason_Name |
|--------------------------|---------|----------|-------------|-------------|
| <input type="checkbox"/> | 1       | 5        | Ready       | Standby     |
| <input type="checkbox"/> | 2       | 60       | Delay       | No Spare    |
| <input type="checkbox"/> | 3       | 10       | Spare       | Cold Spare  |
| <input type="checkbox"/> | 4       | 20       | Down        | Hot Spare   |
| <input type="checkbox"/> | 5       | 15       | Spare       | Hot Spare   |
| <input type="checkbox"/> | 6       | 25       | Delay       | Vendor      |
| <input type="checkbox"/> | 7       | 20       | Ready       | Standby     |
| <input type="checkbox"/> | 8       | 60       | Down        | Broken Part |