



DBMS PROJECT

MSIS 2603 – SPRING 2017

BY: #SCUDATACRATS
KA HOU SIO -W1166793
SHARADA RAMAKRISHNAN - W1118501
TANUSHREE CHAUDHARY - W1361677
YENG HENG PENG -W0873413

TABLE OF CONTENTS

BUSINESS APPLICATION	3
USER TYPES	4
USER CASES	5
ENUMERATING QUERIES FOR USE CASES.....	6
BUSINESS METRICS.....	7
UML.....	10
DATA DICTIONARY	14
CREATION OF DATABASE AND TABLES DATA POPULATION	18
USE CASE QUERY RESULTS.....	19
BUSINESS METRIC QUERIES RESULTS AND REPORTS	34

BUSINESS APPLICATION

Human resource management is designed to increase employee performance aligning with the employer's strategic objectives. It was initially dominated transactional work such as Payroll, Benefits and administration. However due to globalization company consolidation, technological advances and further research, Human resource management has evolved and now focuses on strategic initiatives like mergers and acquisitions, talent management, succession planning, industrial and labor relations and diversity with inclusion.



Objectives of Human Resource Management:

The application mainly provides the company with the following objectives:

- Global perspective to be a major player by transforming and standardizing global practices
- Help employees and managers to be accountable and proactive
- Provides visibility into the workforce planning to manage human capital by business divisions and subsidiaries
- Most importantly analyzing the employee data helps to make informed data driven decisions.

Human resource is a very big domain consisting of various functions like payroll management, recruitment management, performance management as listed in the below diagram. Each function has its own set of functions, users and policies.



The scope of our project is limited to the function of Human Resources Employee Management. Our application is designed to help a company to manage and analyze Employee data in an efficient and effective technique. The main goals of our application are:

- Ease of access through reducing manual steps for HR and employee record management
- Achieving data accuracy by eliminating duplicate entries
- Capability to track and access important metrics and reports
- Improving audit and compliance management

USER TYPES

In our HR management application, we have the following user types:

- **EMPLOYEE** - all employees in the company.
- **HR GENERALIST** - employee that is responsible for human resources related tasks for the company.

- **MANAGER** - there is Vice President, Department Manager, Project Manager and they are responsible for overseeing employee in the company, department and project.
- **APPLICATION ADMINISTRATOR** - oversee the whole application and responsible to perform changes to the application upon requested.
- **HR ANALYST** - Business Analyst that is specialized in human resource sector, generate business metrics/reports and provide insights to management from data.

USER CASES

Based on our assumptions (referenced in exhibits) and users of our application, a list of use cases for each user is as follows:

Employee

1. Upon situation, employee will update personal info such as address, email, phone number and etc.
2. Update/Insert the Employee Skills table once the employee got a new talent

HR Generalist

1. Create a new role and role description for new employee
2. Upon hiring a new employee, insert new employee data
3. Upon termination of an employee, change employee status and update employee date of leaving
4. Upon request, transfer an employee to another department
5. Upon request, change the manager department or project
6. View and audit reports from application administrator

Manager

1. Request to change salary for subordinates upon situation such as promotion, demotion, and etc.
2. Request to promote/demote subordinate

Application Administrator

1. Upon request, will give, change or revoke employee roles for related access to the application
2. Upon request create entry for Project and Department
3. Will generate audit reports for data correctness

HR Analyst

1. Use employee's data to perform analysis and check data health for generation of business metrics report
2. Based on the analysis, will generate metrics report with insights about employee performance for management to view

ENUMERATING QUERIES FOR USE CASES

HR-Generalist

- Create a new job
- Hire a new employee
- Terminate an employee
- Transfer an employee
- Change manager

Employee

- Change Personal information

- Update talent portfolio

Manager

- Request salary changes for subordinates
- Request promotion/demotion for subordinates

Application Administrator

- Give/change/revoke specific roles for related access
- Upon request, will create new entry for jobs, projects and departments
- Provide audit reports

HR Analyst

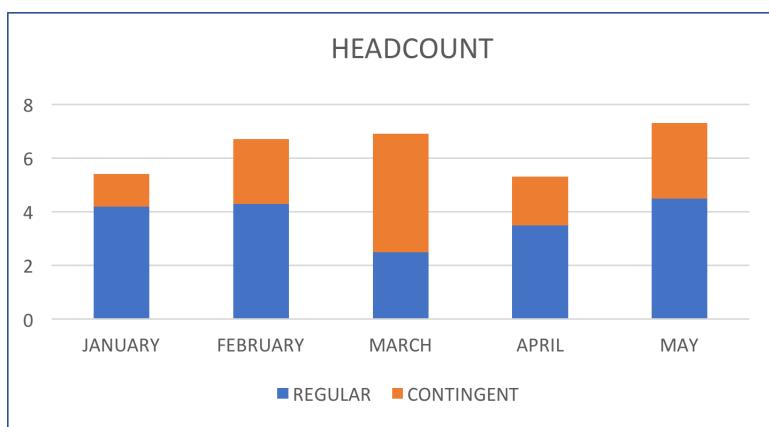
- Use employees' data to perform analysis and check data health
- Generate different types of business metrics

BUSINESS METRICS

The business metrics for our application are as follows:

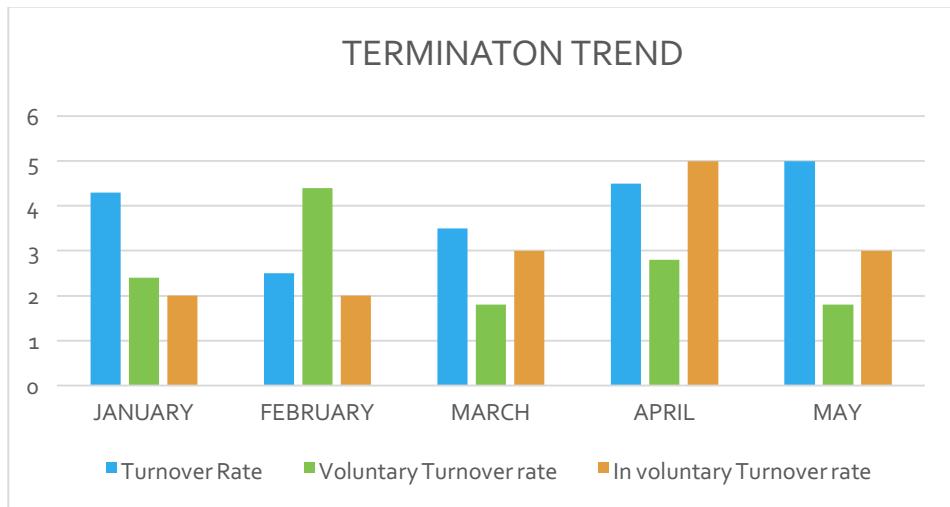
- **Headcount Trend:**

- **Definition:** Overview of number of employees in the workforce over a period of time.
- **Sample Graph type:** Bar graph



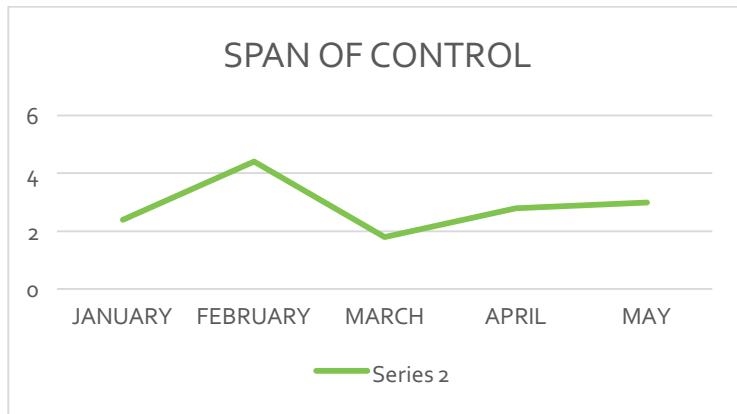
➤ Termination Trend

- **Definition:** % of employees in a workforce who leave during a certain period of time. This can be measured as voluntary vs involuntary turnover rate. For the scope of this project we are considering only the overall turnover rate.
- **Sample Graph type:** Clustered column



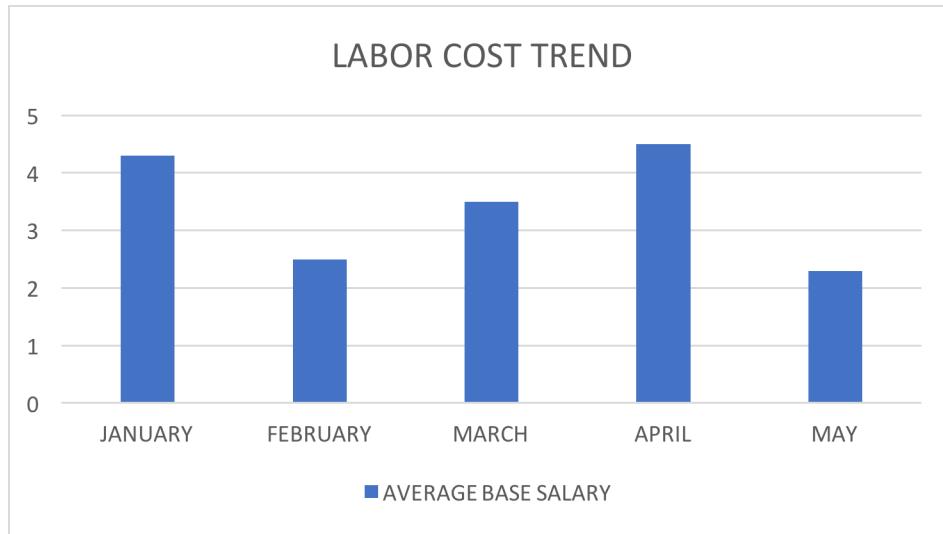
➤ Span of Control

- **Definition:** The number of subordinates under one supervisor.
- **Sample Graph type:** Line Graph



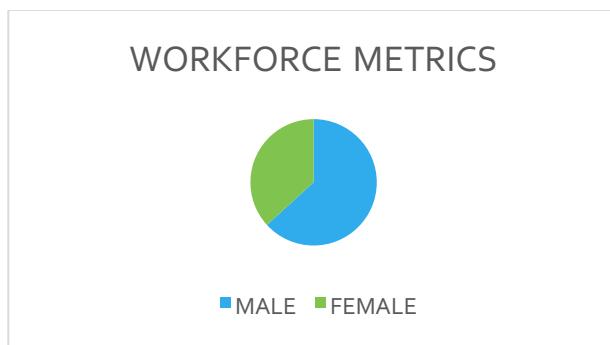
➤ Labor Cost Trend

- **Definition:** Overview of total sum of annual base pay over time for the entire headcount.
- **Sample Graph type:** Bar Graph



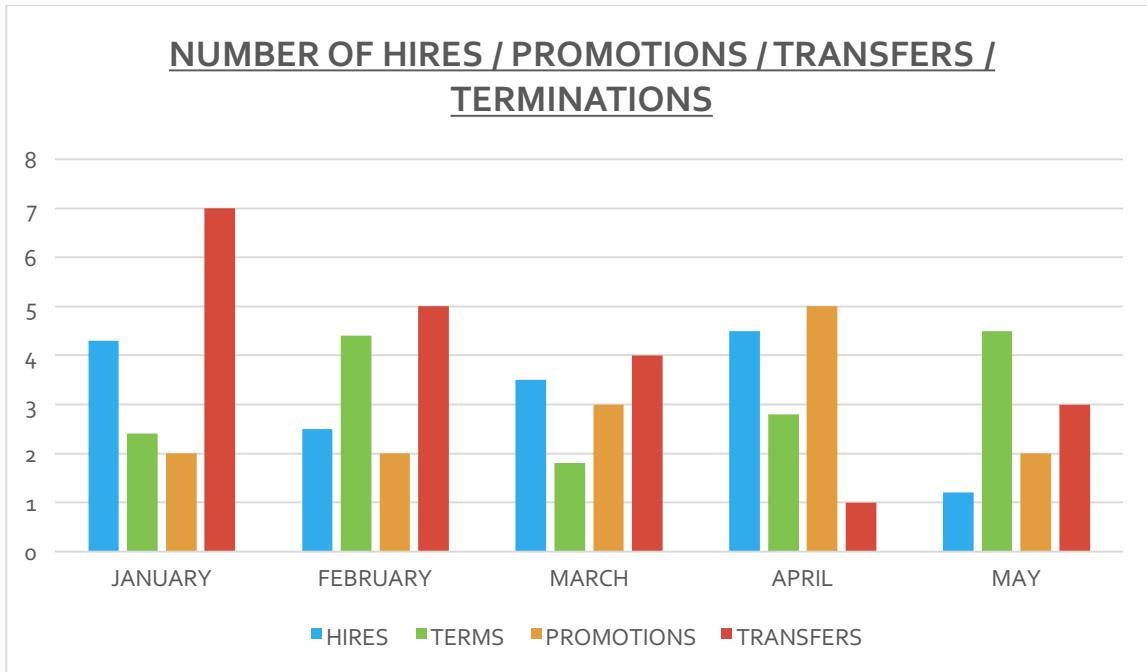
➤ Workforce Mix Metrics

- **Definition:** There can be different time of metrics like generation, tenure, hire, gender and talent. For the scope of this project we are considering only the gender workforce metric.
- **Sample Graph type:** Pie chart



➤ **Number of Promotions, Hires, Terms, Transfers**

- **Definition:** The number of Hires, Terms, Transfers and Promotions over a period of time.
- **Sample Graph type:** Clustered column



UML

Main Entities and Assumptions:

Employee:

- An Employee works under one Project Manager only
- An Employee works for one Department only
- An Employee can work in one or more Projects if the projects are under same Manager and same Department

- Employee status can be Active, Inactive, STL (Short Term Leave 1-3 days leave), LTL (Long Term Leave 4-15 days leave) depending on the request type –Time off.

Role:

- One Employee can have one Access Role only
- Access Roles can be Department Manager, Project Manager, Vice President, HR Generalist, HR Analyst, Admin and Employee only

Department:

- One Department can be under one Location only
- One Department can have multiple Projects
- One Department can have one Department manager only

Location:

- One Location can have multiple Departments

Project:

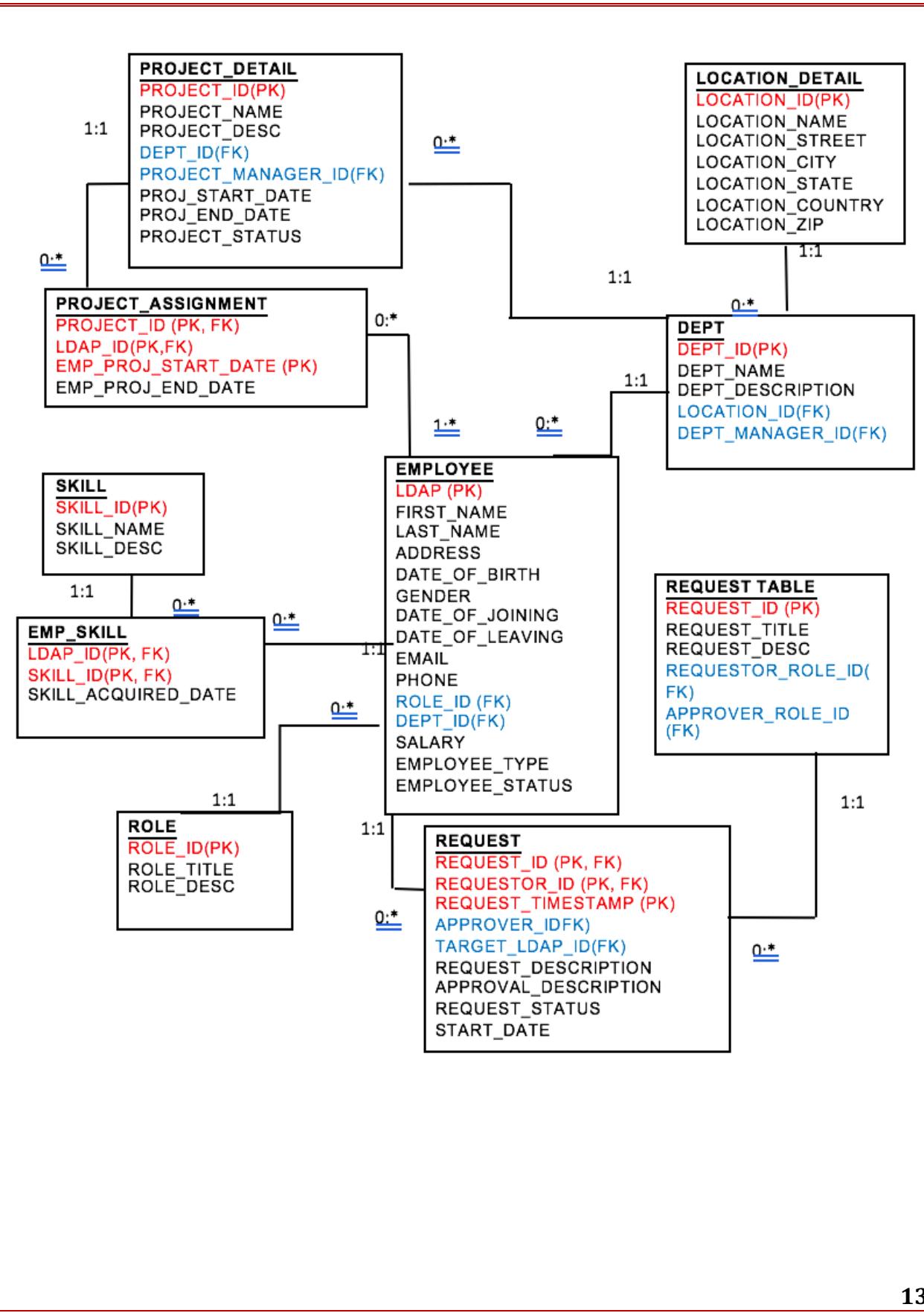
- One project can be assigned to one or more Employees
- Date on which the employee joins a project is unique
- One project can be owned by one project manager only
- One project Manager can handle one or more Projects, one or more employees under the same department
- Project status can be On-going, Completed.

Skill:

- One Employee can have zero or more Skills
- Skills are defined in the organization skill list with a name and description

Request:

- One Request can have one Approver only
- One Employee can make multiple Requests and same requests at different times of the day
- Approver can approve multiple requests
- Requests are defined in the organization request list with a name, description, type, requestor role and approver role.
- Based on the request, the requestor role and approver role will be pre-defined for the process flow in the request table.
- Request status can be Initiated, Cancelled, Completed, On-going (more than 15 days leave)
- The Leave policy specifies that if leave is STL – 1 to 3 days leave, Vacation – 4 to 15 days, LTL – up to 40 days, beyond 40 days leave its Inactive.



DATA DICTIONARY

ROLE
ROLE_ID: VARCHAR (15) (AUTO) // PRIMARY KEY
ROLE_TITLE: VARCHAR (30) NOT NULL
ROLE_DESCRIPTION: VARCHAR (50)

EMPLOYEE
LDAP: VARCHAR (15) (AUTO) // PRIMARY KEY
FIRST_NAME: VARCHAR (30) NOT NULL
LAST_NAME: VARCHAR (30) NOT NULL
STREET: VARCHAR (50)
CITY: VARCHAR (50)
ZIP: INT (50) NOT NULL
STATE: VARCHAR (50)
COUNTRY: VARCHAR (50)
DATE_OF_BIRTH: DATETIME NOT NULL
GENDER: VARCHAR (10) NOT NULL
DATE_OF_JOINING: DATETIME
DATE_OF_LEAVE: DATETIME
PERSONAL_EMAIL: VARCHAR (50) NOT NULL
WORK_EMAIL: VARCHAR (50) NOT NULL

PERSONAL_PHONE: VARCHAR (50) NOT NULL
WORK_PHONE: VARCHAR (50)
ROLE_ID: VARCHAR (15) NOT NULL // FOREIGN KEY TO ROLE (ROLE_ID)
DEPT_ID: VARCHAR (15) NOT NULL // FOREIGN KEY TO DEPARTMENT (DEPT_ID)
BASE_SALARY: FLOAT NOT NULL
EMPLOYEE_TYPE: VARCHAR (50)
EMPLOYEE_STATUS: VARCHAR (30) // DEFAULT 'ACTIVE' – {ACTIVE, INACTIVE, STL, LTL, VACATION}

LOCATION_DETAIL
LOCATION_ID: VARCHAR (15)// PRIMARY KEY
LOCATION_NAME: VARCHAR (10) NOT NULL
LOCATION_STREET: VARCHAR (10) NOT NULL
LOCATION_STATE: VARCHAR (10) NOT NULL
LOCATION_CITY: VARCHAR (10) NOT NULL
LOCATION_COUNTRY: VARCHAR (10) NOT NULL
LOCATION_ZIP: VARCHAR (20) NOT NULL

SKILL
SKILL_ID: VARCHAR (15) //PRIMARY KEY
SKILL_NAME: VARCHAR (30) NOT NULL

DEPARTMENT
DEPT_ID: VARCHAR (15) // PRIMARY KEY
DEPT_NAME: VARCHAR (30) NOT NULL
DEPT_DESCRIPTION: VARCHAR (50)
DEPT_MANAGER_ID: VARCHAR (15) // FOREIGN KEY TO EMPLOYEE (LDAP)
LOCATION_ID: VARCHAR (15) // FOREIGN KEY TO LOCATION_DETAIL (LOCATION_ID)
SKILL_DESCRIPTION: VARCHAR (50)

EMP_SKILL
LDAP_ID: VARCHAR (15) // PRIMARY KEY, FOREIGN KEY TO EMPLOYEE(LDAP)
SKILL_ID: VARCHAR (15) // PRIMARY KEY, FOREIGN KEY TO SKILL(SKILL_ID)
SKILL_DESCRIPTION: VARCHAR (50)

PROJECT_ASSIGNMENT
PROJECT_ID: VARCHAR (15) // PRIMARY KEY, FOREIGN KEY TO PROJECT_DETAIL(PROJECT_ID)
LDAP_ID: VARCHAR (15) // PRIMARY KEY, FOREIGN KEY TO EMPLOYEE (LDAP)

EMP_PROJ_START_DATE: DATETIME // PRIMARY KEY

PROJECT_DETAIL

PROJECT_ID: VARCHAR (15) // PRIMARY KEY

PROJECT_NAME: VARCHAR (30) NOT NULL

DESCRIPTION: VARCHAR (50)

DEPT_ID: VARCHAR (15) NOT NULL // FOREIGN KEY TO DEPARTMENT
(DEPT_ID)

PROJECT_MANAGER_ID: VARCHAR (15) NOT NULL// FOREIGN KEY TO
EMPLOYEE(LDAP)

START_DATE: DATETIME NOT NULL

END_DATE: DATETIME

PROJECT_STATUS: VARCHAR (30) NOT NULL// DEFAULT 'ON-GOING' - {ON-GOING, COMPLETED}

REQUEST_TABLE

REQUEST_ID: VARCHAR (15) NOT NULL// PRIMARY KEY

REQUEST_TITLE: VARCHAR (30) NOT NULL

REQUEST_DESCRIPTION: VARCHAR (50) NOT NULL

REQUESTOR_ROLE_ID: VARCHAR (15) NOT NULL // FOREIGN KEY TO
EMPLOYEE (LDAP)

APPROVER_ROLE_ID: VARCHAR (15) NOT NULL // FOREIGN KEY TO

EMPLOYEE (LDAP)

REQUEST

REQUEST_ID: VARCHAR (30) NOT NULL // PRIMARY KEY, FOREIGN KEY TO REQUEST_TABLE(REQUEST_ID)

REQUESTOR_ID: VARCHAR (15) NOT NULL // PRIMARY KEY, FOREIGN KEY TO EMPLOYEE (LDAP)

REQUEST_TIMESTAMP: DATETIME NOT NULL // PRIMARY KEY

APPROVER_ID: VARCHAR (15) NOT NULL // FOREIGN KEY TO EMPLOYEE (LDAP)

REQUEST_DESCRIPTION: VARCHAR (200)

APPROVAL_DESCRIPTION: VARCHAR (200)

REQUEST_STATUS: VARCHAR (10) NOT NULL //DEFAULT –‘ONGOING’

START_DATE: DATETIME NOT NULL

END_DATE: DATETIME

TARGET_LDAP_ID VARCHAR (15) // FOREIGN KEY TO EMPLOYEE (LDAP)

CREATION OF DATABASE AND TABLES DATA POPULATION

The creation and insertion of data is present in the SQL file #SCUDATACRATS DBMS PROJECT .sql file.

USE CASE QUERY RESULTS

HR-Generalist

- Create a new job

Before:

SELECT * FROM ROLE

The screenshot shows a database query results grid. At the top, there is a code editor with the following SQL query:

```
1 SELECT *
2 FROM ROLE
```

Below the code editor is a grid view showing the results of the query. The grid has columns: ROLE_ID, ROLE_TITLE, and ROLE_DESCRIPTION. There are 7 rows of data:

ROLE_ID	ROLE_TITLE	ROLE_DESCRIPTION
1 R1	VICE PRESIDENT	Oversee and manage all departments and responsible for everything inside the department
2 R2	DEPARTMENT MANAGER	Manager in a department, subordinate might have employee, HR generalist, Manager, HR analy...
3 R3	PROJECT MANAGER	Responsible for managing the project workflow per project requirement. Need to balance const...
4 R4	HR GENERALIST	Oversee company human resources, tasks such as requesting, checking employee's info
5 R5	HR ANALYST	Responsible for analyzing company HR data, and create dash board and insights from those da...
6 R6	APPLICATION ADMIN	Oversee the whole HR application, responsible to checking data of the application, makes chan...
7 R7	EMPLOYEE	Employee of the company, work under employee

After:

INSERT INTO ROLE ('ROLE_ID', 'ROLE_TITLE', 'ROLE_DESCRIPTION') VALUES
(‘R8’, ‘ACCOUNTANT’, ‘To manage company or department finance related tasks’)

SELECT * FROM ROLE

The screenshot shows a database query results grid. At the top, there is a code editor with the following SQL query:

```
1 INSERT INTO ROLE ('ROLE_ID', 'ROLE_TITLE', 'ROLE_DESCRIPTION')
2 VALUES ('R8','ACCOUNTANT','To manage company or department finance related tasks')
3
4 SELECT *
5 FROM ROLE
```

Below the code editor is a grid view showing the results of the query. The grid has columns: ROLE_ID, ROLE_TITLE, and ROLE_DESCRIPTION. There are 8 rows of data:

ROLE_ID	ROLE_TITLE	ROLE_DESCRIPTION
1 R1	VICE PRESIDENT	Oversee and manage all departments and responsible for everything inside the department
2 R2	DEPARTMENT MANAGER	Manager in a department, subordinate might have employee, HR generalist, Manager, HR analy...
3 R3	PROJECT MANAGER	Responsible for managing the project workflow per project requirement. Need to balance const...
4 R4	HR GENERALIST	Oversee company human resources, tasks such as requesting, checking employee's info
5 R5	HR ANALYST	Responsible for analyzing company HR data, and create dash board and insights from those da...
6 R6	APPLICATION ADMIN	Oversee the whole HR application, responsible to checking data of the application, makes chan...
7 R7	EMPLOYEE	Employee of the company, work under employee
8 R8	ACCOUNTANT	To manage company or department finance related tasks

- Hire a new employee

Before:

```
SELECT * FROM EMPLOYEE
```

	LDAP	FIRST_NAME	LAST_NAME	STREET	CITY	ZIP	STATE	COUNTRY	DATE_OF_BIRTH	GENDER	DATE_OF_JOINING
7	7	STEVE	JOBSS	EAST PALO ST	MORDOR	45673	NEVADA	USA	1970-07-20	MALE	2010-03-02
8	8	DIANA	ROSS	FRANKLIN ST	CAPITAL CITY	73742	NORTH CAROLINA	USA	1975-09-21	FEMALE	2011-06-19
9	9	MARY	STEVENS	BENTON ST	CAMELOT	76483	CALIFORNIA	USA	1965-10-30	FEMALE	2013-04-23
10	10	DIANA	SELLER	SHERMEN ST	NARNIA	27474	NEW JERSEY	USA	1998-02-27	FEMALE	2017-01-29
11	11	DEBORAH	HAYNES	SOUTH COUNTRY CLUB ROAD	COVINGTON	30014	GEORGIA	USA	1989-03-31	FEMALE	2011-07-29
12	12	MOSES	GARRETT	STATE ST	WILLIAMSBURG	23195	VIRGINIA	USA	2002-02-22	FEMALE	2011-05-29
13	13	LLYOD	AUSTIN	ROOSEVELT DR	WEST BEND	53095	WISCONSIN	USA	1995-05-22	FEMALE	2015-03-29
14	14	LEROY	NEWTON	CROSS ST	MAYS LANDING	8330	NEW JERSEY	USA	1977-11-14	FEMALE	2005-11-29
15	15	MICHELLE	TURNER	RIVERVIEW AVE	NORTH AUGUSTA	29841	SOUTH CAROLINA	USA	1984-07-26	FEMALE	2006-12-01
16	16	MITCHELL	SCHWARTZ	PROSPECT AVE	MC LEAN	22101	VIRGINIA	USA	1994-01-25	FEMALE	2016-02-15
17	17	PAT	FLOYD	RIVER DR	SOLON	44139	OHIO	USA	1998-05-10	FEMALE	2017-01-15
18	18	RAY	BISHOP	NE ORCHARD ST	MADISON HEIGHTS	48071	MICHIGAN	USA	1968-02-21	MALE	2005-04-29
19	19	ELENA	GEORGE	MAYFLOWER RD	SOUTH WINDSOR	6074	CONNECTICUT	USA	1988-05-13	FEMALE	2006-01-03
20	20	BRADFORD	MILLER	DELAWARE DR	POINT PLEASANT BEACH	8742	NEW JERSEY	USA	1993-05-19	MALE	2016-08-01

After:

```
INSERT INTO EMPLOYEE ('LDAP', 'FIRST_NAME', 'LAST_NAME',
'STREET','CITY','ZIP','STATE','COUNTRY','DATE_OF_BIRTH','GENDER','DATE_OF_JOINING','DATE_OF_LEAVE','PERSONAL_EMAIL','WORK_EMAIL',
'PERSONAL_PHONE', 'WORK_PHONE', 'ROLE_ID', 'BASE_SALARY', 'DEPT_ID',
'EMPLOYEE_TYPE', 'EMPLOYEE_STATUS') VALUES ('21', 'APPLE', 'NEWTON', 'LEO
DR', 'SAN JOSE', '95129', 'CALIFORNIA', 'USA', '1990-04-23', 'FEMALE', '2015-03-01',
NULL, 'APPLE_N@GMAIL.COM', 'APPLE_N@CSKT.COM', '1-408-123-4567', '1-408-543-2365', 'R8', '$75000.00', 'D2', 'REGULAR', 'ACTIVE')
```

SELECT * FROM EMPLOYEE

```
1 INSERT INTO EMPLOYEE ('LDAP', 'FIRST_NAME', 'LAST_NAME', 'STREET', 'CITY', 'ZIP', 'STATE', 'COUNTRY', 'DATE_OF_BIRTH', 'GENDER', 'DATE_OF_JOINING',
2 'DATE_OF_LEAVEING', 'PERSONAL_EMAIL', 'WORK_EMAIL', 'PERSONAL_PHONE', 'WORK_PHONE', 'ROLE_ID', 'BASE_SALARY', 'DEPT_ID', 'EMPLOYEE_TYPE'
3 , 'EMPLOYEE_STATUS')
4 VALUES ('21', 'APPLE', 'NEWTON', 'LEO DR', 'SAN
5 JOSE', '95129', 'CALIFORNIA', 'USA', '1990-04-23', 'FEMALE', '2015-03-01', NULL, 'APPLE_N@GMAIL.COM', 'APPLE_N@CSKT.COM', '1-408-123-4567', '1-408-543-2365',
6 'R8', '$75000.00', 'D2', 'REGULAR', 'ACTIVE')
7
8 SELECT *
9 FROM EMPLOYEE
```

Employee Data										
LDAP	FIRST_NAME	LAST_NAME	STREET	CITY	ZIP	STATE	COUNTRY	DATE_OF_BIRTH	GENDER	DATE_OF_JOINING
8	8	DIANA	ROSS	FRANKLIN ST	CAPITAL CITY	73742	NORTH CAROLINA	USA	1975-09-21	FEMALE 2011-06-19
9	9	MARY	STEVENS	BENTON ST	CAMELOT	76483	CALIFORNIA	USA	1965-10-30	FEMALE 2013-04-23
10	10	DIANA	SELLER	SHERMEN ST	NARNIA	27474	NEW JERSEY	USA	1998-02-27	FEMALE 2017-01-29
11	11	DEBORAH	HAYNES	SOUTH COUNTRY CLUB ROAD	COVINGTON	30014	GEORGIA	USA	1989-03-31	FEMALE 2011-07-29
12	12	MOSES	GARRETT	STATE ST	WILLIAMSBURG	23195	VIRGINIA	USA	2002-02-22	FEMALE 2011-05-29
13	13	LLOYD	AUSTIN	ROOSEVELT DR	WEST BEND	53095	WISCONSIN	USA	1995-05-22	FEMALE 2015-03-29
14	14	LEROY	NEWTON	CROSS ST	MAYS LANDING	8330	NEW JERSEY	USA	1977-11-14	FEMALE 2005-11-29
15	15	MICHELLE	TURNER	RIVERVIEW AVE	NORTH AUGUSTA	29841	SOUTH CAROLINA	USA	1984-07-26	FEMALE 2006-12-01
16	16	MITCHELL	SCHWARTZ	PROSPECT AVE	MC LEAN	22101	VIRGINIA	USA	1994-01-25	FEMALE 2016-02-15
17	17	PAT	FLOYD	RIVER DR	SOLON	44139	OHIO	USA	1998-05-10	FEMALE 2017-01-15
18	18	RAY	BISHOP	NE ORCHARD ST	MADISON HEIGHTS	48071	MICHIGAN	USA	1968-02-21	MALE 2005-04-29
19	19	ELENA	GEORGE	MAYFLOWER RD	SOUTH WINDSOR	6074	CONNECTICUT	USA	1988-05-13	FEMALE 2006-01-03
20	20	BRADFORD	MILLER	DELAWARE DR	POINT PLEASANT BEACH	8742	NEW JERSEY	USA	1993-05-19	MALE 2016-08-01
21	21	APPLE	NEWTON	LEO DR	SAN JOSE	95129	CALIFORNIA	USA	1990-04-23	FEMALE 2015-03-01

- Terminate an employee

Before:

```
SELECT * FROM EMPLOYEE WHERE LDAP = '21'
```

Employee Data Analysis											
Employee Details		Personal Information		Work Contact		Role & Salary		Department		Employment Status	
Employee ID	Last Name	First Name	Email	Phone	Address	Role	Base Salary	Dept ID	Dept Name	Employee Type	Status
1	SELECT *										
2	FROM EMPLOYEE										
Total rows loaded: 21											
Grid view Form view											
3	NULL	DIANA	RIGGMAIL.COM	DIANA.RIGGS@SKT.COM	1-916-555-1000	1-408-543-2352	\$4	\$7342.00	D2	REGULAR	LTL
4	NULL	MARY	SIMPSONMAIL.COM	MARY.SIMPSON@SKT.COM	1-405-555-1277	1-408-543-2353	\$4	\$234,356.00	D3	REGULAR	ACTIVE
5	NULL	DIANA	GIGGMAIL.COM	DIANA.GIGGS@SKT.COM	1-334-555-1400	1-408-543-2354	\$5	\$103,496.00	D2	REGULAR	ACTIVE
6	NULL	DEBORAH	HGOMAL.COM	DEBORAH.HGOMA@SKT.COM	1-518-555-0194	1-408-543-2355	\$5	\$109,665.00	D3	REGULAR	ACTIVE
7	NULL	MOSES	GIGGMAIL.COM	MOSES.GIGGS@SKT.COM	1-518-555-0196	1-408-543-2356	\$6	\$158,922.00	D2	REGULAR	STL
8	NULL	LLYOD	AIGMAIL.COM	LLYOD.AIG@SKT.COM	1-225-555-0188	1-408-543-2357	\$6	\$141,836.00	D3	REGULAR	INACTIVE
9	NULL	LEROY	NIGMAIL.COM	LEROY.NIG@SKT.COM	1-701-555-0118	1-408-543-2358	\$7	\$125,189.00	D2	REGULAR	ACTIVE
10	NULL	MICHELLE	TIGMAIL.COM	MICHELLE.TIG@SKT.COM	1-208-555-0195	1-408-543-2359	\$7	\$162,554.00	D2	REGULAR	ACTIVE
11	5-01	MITCHEL	LGIGMAIL.COM	MITCHEL.LGIG@SKT.COM	1-775-555-0192	1-408-543-2360	\$7	\$54,105.00	D2	REGULAR	TERMINATED
12	NULL	PAT	FGBMAIL.COM	PAT.FGB@SKT.COM	1-815-555-0138	1-408-543-2361	\$7	\$59,596.00	D2	CONTINGENT	VACATION
13	NULL	RAY	BGBMAIL.COM	RAY.BGB@SKT.COM	1-512-555-0121	1-408-543-2362	\$7	\$193,921.00	D3	REGULAR	ACTIVE
14	NULL	ELENA	GIGMAIL.COM	ELENA.GIG@SKT.COM	1-609-555-0162	1-408-543-2363	\$7	\$37,368.00	D3	REGULAR	ACTIVE
15	NULL	BRADFORD	MIGMAIL.COM	BRADFORD.MIG@SKT.COM	1-505-555-0132	1-408-543-2364	\$7	\$169,903.00	D3	REGULAR	ACTIVE
16	NULL	APPLE	NGMAIL.COM	APPLE.NG@SKT.COM	1-408-123-4567	1-408-543-2365	\$8	\$7500.00	D2	REGULAR	ACTIVE

After:

```
UPDATE EMPLOYEE SET DATE_OF_LEAVEING = '2017-06-05', EMPLOYEE_STATUS =  
'TERMINATED' WHERE LDAP = '21'
```

```
SELECT * FROM EMPLOYEE WHERE LDAP = '21'
```

The screenshot shows a database grid interface. At the top, there is a code editor window containing the following SQL statements:

```
1 UPDATE EMPLOYEE  
2 SET DATE_OF_LEAVEING = '2017-06-05', EMPLOYEE_STATUS = 'TERMINATED'  
3 WHERE LDAP = '21'  
4  
5 SELECT *  
6 FROM EMPLOYEE  
7 WHERE LDAP = '21'
```

Below the code editor is a toolbar with various icons. To its right, a message says "Total rows loaded: 1". The main area displays a single row of data in a table format:

	NAME	DATE_OF_LEAVEING	PERSONAL_EMAIL	WORK_EMAIL	PERSONAL_PHONE	WORK_PHONE	ROLE_ID	BASE_SALARY	DEPT_ID	EMPLOYEE_TYPE	EMPLOYEE_STATUS
1		2017-06-05	APPLE_N@GMAIL.COM	APPLE_N@CSKT.COM	1-408-123-4567	1-408-543-2365	R8	\$75000.00	D2	REGULAR	TERMINATED

- Transfer an employee

Before:

```
SELECT DEPT_ID FROM EMPLOYEE WHERE LDAP = '21'
```

The screenshot shows a database grid interface. At the top, there is a code editor window containing the following SQL statement:

```
1 SELECT DEPT_ID  
2 FROM EMPLOYEE  
3 WHERE LDAP = '21'  
4
```

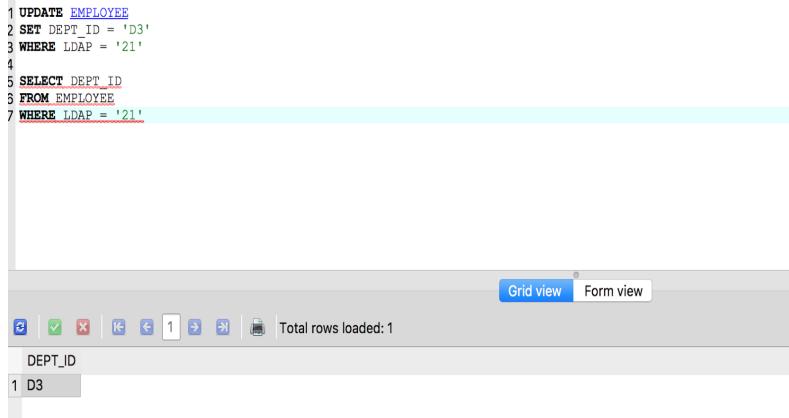
Below the code editor is a toolbar with various icons. To its right, a message says "Total rows loaded: 1". The main area displays a single row of data in a table format:

DEPT_ID
1 D2

After:

```
UPDATE EMPLOYEE SET DEPT_ID = 'D3' WHERE LDAP = '21'
```

```
SELECT DEPT_ID FROM EMPLOYEE WHERE LDAP = '21'
```



The screenshot shows the Oracle SQL Developer interface. The code editor at the top contains the following SQL statements:

```
1 UPDATE EMPLOYEE
2 SET DEPT_ID = 'D3'
3 WHERE LDAP = '21'
4
5 SELECT DEPT_ID
6 FROM EMPLOYEE
7 WHERE LDAP = '21'
```

The result set below shows one row with DEPT_ID 'D3'.

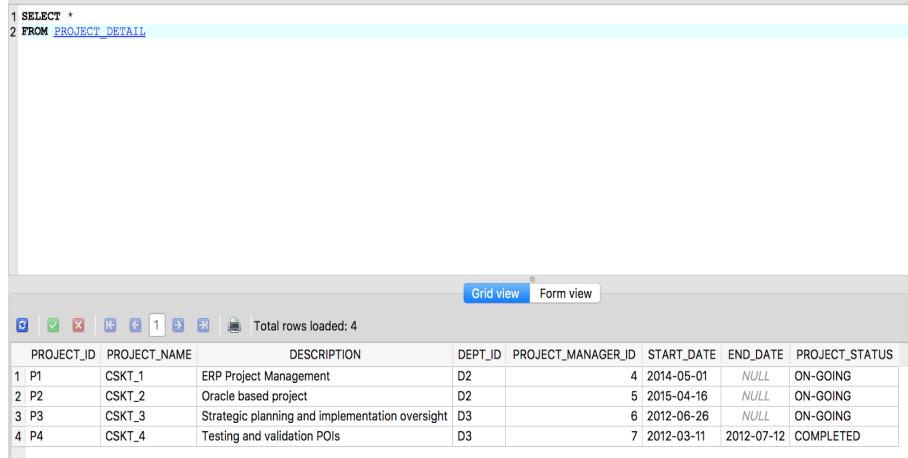
DEPT_ID
1 D3

➤ Change manager

a) For Project Manager

Before:

```
SELECT * FROM PROJECT_DETAIL
```



The screenshot shows the Oracle SQL Developer interface. The code editor at the top contains the following SQL statement:

```
1 SELECT *
2 FROM PROJECT_DETAIL
```

The result set below shows four rows of project details.

PROJECT_ID	PROJECT_NAME	DESCRIPTION	DEPT_ID	PROJECT_MANAGER_ID	START_DATE	END_DATE	PROJECT_STATUS
1 P1	CSKT_1	ERP Project Management	D2	4	2014-05-01	NULL	ON-GOING
2 P2	CSKT_2	Oracle based project	D2	5	2015-04-16	NULL	ON-GOING
3 P3	CSKT_3	Strategic planning and implementation oversight	D3	6	2012-06-26	NULL	ON-GOING
4 P4	CSKT_4	Testing and validation POIs	D3	7	2012-03-11	2012-07-12	COMPLETED

After:

```
UPDATE PROJECT_DETAIL SET PROJECT_MANAGER_ID = '5' WHERE
PROJECT_ID = 'P1' SELECT * FROM PROJECT_DETAIL
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL code:

```
1 UPDATE PROJECT_DETAIL
2 SET PROJECT_MANAGER_ID = '5'
3 WHERE PROJECT_ID = 'P1'
4
5 SELECT *
6 FROM PROJECT_DETAIL
7
```

In the bottom-right pane, there is a data grid titled "Grid view" showing the results of the query. The results are as follows:

PROJECT_ID	PROJECT_NAME	DESCRIPTION	DEPT_ID	PROJECT_MANAGER_ID	START_DATE	END_DATE	PROJECT_STATUS
1 P1	CSKT_1	ERP Project Management	D2	5	2014-05-01	NULL	ON-GOING
2 P2	CSKT_2	Oracle based project	D2	5	2015-04-16	NULL	ON-GOING
3 P3	CSKT_3	Strategic planning and implementation oversight	D3	6	2012-06-26	NULL	ON-GOING
4 P4	CSKT_4	Testing and validation POIs	D3	7	2012-03-11	2012-07-12	COMPLETED

b) For Department Manager

Before:

```
SELECT * FROM DEPARTMENT
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL code:

```
1 SELECT *
2 FROM DEPARTMENT
3
```

In the bottom-right pane, there is a data grid titled "Grid view" showing the results of the query. The results are as follows:

DEPT_ID	DEPT_NAME	DEPT_DESCRIPTION	LOCATION_ID	DEPT_MANAGER_ID
1 D1	Organization	Headquarters	L1	1
2 D2	Finance	Records and reports the cash flow transactions of the company	L2	2
3 D3	Technology	Investigate activities a business conducts to improve existing products and procedure	L3	3

After:

```
UPDATE DEPARTMENT SET DEPT_MANAGER_ID = '1' WHERE DEPT_ID = 'D2'
```

SELECT * FROM DEPARTMENT

```
1 UPDATE DEPARTMENT
2 SET DEPT_MANAGER_ID = '1'
3 WHERE DEPT_ID = 'D2'
4
5 SELECT *
6 FROM DEPARTMENT
```

Grid view Form view

Total rows loaded: 3

DEPT_ID	DEPT_NAME	DEPT_DESCRIPTION	LOCATION_ID	DEPT_MANAGER_ID	
1	D1	Organization	Headquarters	L1	1
2	D2	Finance	Records and reports the cash flow transactions of the company	L2	1
3	D3	Technology	Investigate activities a business conducts to improve existing products and procedure	L3	3

Employee

- Change Personal information

Before:

```
SELECT LDAP, FIRST_NAME, LAST_NAME, STREET, CITY, ZIP, STATE, COUNTRY,
PERSONAL_EMAIL, PERSONAL_PHONE, BASE_SALARY
FROM EMPLOYEE
WHERE LDAP = '1'
```

```
1 SELECT LDAP, FIRST_NAME, LAST_NAME, STREET, CITY, ZIP, STATE, COUNTRY, PERSONAL_EMAIL, PERSONAL_PHONE, BASE_SALARY
2 FROM EMPLOYEE
3 WHERE LDAP = '1'
```

Grid view Form view

Total rows loaded: 1

LDAP	FIRST_NAME	LAST_NAME	STREET	CITY	ZIP	STATE	COUNTRY	PERSONAL_EMAIL	PERSONAL_PHONE	BASE_SALARY	
1	1	BARBARA	GORDEN	CIVIC CENTER DR	GOTHAM	95123	CALIFORNIA	USA	BARBARA_G@GMAIL.COM	1-601-555-164	\$329,631.00

After:

```
UPDATE EMPLOYEE
```

```

SET STREET = 'EL CAMINO REAL', CITY = 'SANTA CLARA', ZIP = '95054', STATE =
'CALIFORNIA', COUNTRY = 'USA', PERSONAL_EMAIL =
'BARBARA_G@HOTMAIL.COM', PERSONAL_PHONE = '1-408-874-7463',
BASE_SALARY = '400,000.00'

WHERE LDAP = '1'

```

```

SELECT FIRST_NAME, LAST_NAME, STREET, CITY, ZIP, STATE, COUNTRY,
PERSONAL_EMAIL, PERSONAL_PHONE, BASE_SALARY
FROM EMPLOYEE

```

WHERE LDAP = '1'

➤ Update talent portfolio

a) New skill category for the company talent pool

Before:

```
SELECT * FROM SKILL
```

The screenshot shows a database interface with a SQL query editor at the top and a results grid below. The query is:

```

1 SELECT *
2 FROM SKILL
3 |

```

The results grid has columns: SKILL_ID, SKILL_NAME, and SKILL_DESCRIPTION. The data is as follows:

	SKILL_ID	SKILL_NAME	SKILL_DESCRIPTION
1	S1	ACCOUNTING	SKILL BASED ON GAAP ACCOUNTING RULES, CORPORATE ACCOUNTING AND CONTROL
2	S2	MARKETING	SKILL BASED ON PRODUCT, TECHNICAL, CHANNEL, SOCIAL MEDIA OR MOBILE MARKETING
3	S3	OPERATIONS MANAGEMENT	SKILL BASED ON SUPPLY CHAIN AND INVENTORY MANAGEMENT
4	S4	PRODUCT DEVELOPMENT	SKILL BASED ON PRODUCT DEVELOPMENT AND LIFECYCLE
5	S5	QUALITY CONTROL	SKILL FOR QUALITY ASSURANCE AND MANAGEMENT
6	S6	DATA ANALYSIS	SKILL BASED ON R, PYTHON, DBMS, HADOOP, FORECASTING, WAREHOUSING
7	S7	NETWORKING	SKILL BASED ON NETWORKING PROTOCOLS, MODELS AND MANAGEMENT
8	S8	PROJECT MANAGEMENT	SKILL BASED ON LEADERSHIP, COMMUNICATION, NEGOTIATIONS
9	S9	FINANCIAL REPORTING	SKILL BASED ON FINANCIAL ANALYSIS

After:

```

INSERT INTO SKILL ('SKILL_ID', 'SKILL_NAME', 'SKILL_DESCRIPTION')
VALUES ('S10', 'COACHING', 'SKILL BASED ON ABAILITY TO TEACH AND
MENTOR ON HIS OR HER TECHNICAL AREA')

SELECT * FROM SKILL

```

```

1 INSERT INTO SKILL ('SKILL_ID', 'SKILL_NAME', 'SKILL_DESCRIPTION')
2 VALUES ('S10', 'COACHING', 'SKILL BASED ON ABAILITY TO TEACH AND
3 MENTOR ON HIS OR HER TECHNICAL AREA')
4 SELECT *
5 FROM SKILL

```

SKILL_ID	SKILL_NAME	SKILL_DESCRIPTION
1	S1	SKILL BASED ON GAAP ACCOUNTING RULES, CORPORATE ACCOUNTING AND CONTROL
2	S2	SKILL BASED ON PRODUCT, TECHNICAL, CHANNEL, SOCIAL MEDIA OR MOBILE MARKETING
3	S3	SKILL BASED ON SUPPLY CHAIN AND INVENTORY MANAGEMENT
4	S4	SKILL BASED ON PRODUCT DEVELOPMENT AND LIFECYCLE
5	S5	SKILL FOR QUALITY ASSURANCE AND MANAGEMENT
6	S6	SKILL BASED ON R, PYTHON, DBMS, HADOOP, FORECASTING, WAREHOUSING
7	S7	SKILL BASED ON NETWORKING PROTOCOLS, MODELS AND MANAGEMENT
8	S8	SKILL BASED ON LEADERSHIP, COMMUNICATION, NEGOTIATIONS
9	S9	SKILL BASED ON FINANCIAL ANALYSIS
10	S10	SKILL BASED ON ABILITY TO TEACH AND MENTOR ON HIS OR HER TECHNICAL AREA

b) New skill an employee obtained

Before:

```
SELECT * FROM EMP_SKILL WHERE LDAP_ID = '1'
```

```

1 SELECT *
2 FROM EMP_SKILL
3 WHERE LDAP_ID = '1'

```

LDAP_ID	SKILL_ID	SKILL_DESCRIPTION	SKILL_ACQUIRED_DATE
1	1	S1	NULL
1	1	S4	2008-08-13

After:

```
INSERT INTO EMP_SKILL ('LDAP_ID', 'SKILL_ID', 'SKILL_ACQUIRED_DATE')  
VALUES ('1', 'S10', '2015-05-01')
```

```
SELECT * FROM EMP_SKILL WHERE LDAP_ID = '1'
```

The screenshot shows a database interface with two panes. The top pane is a 'Query' window containing the following SQL code:

```
1 INSERT INTO EMP_SKILL ('LDAP_ID', 'SKILL_ID', 'SKILL_ACQUIRED_DATE')  
2 VALUES ('1', 'S10', '2015-05-01')  
3  
4 SELECT *  
5 FROM EMP_SKILL  
6 WHERE LDAP_ID = '1'  
7
```

The bottom pane is a 'Grid view' showing the results of the query. The table has columns: LDAP_ID, SKILL_ID, SKILL_DESCRIPTION, and SKILL_ACQUIRED_DATE. The data is as follows:

LDAP_ID	SKILL_ID	SKILL_DESCRIPTION	SKILL_ACQUIRED_DATE
1	S1	NULL	1990-10-21
2	S10	NULL	2015-05-01
3	S4	NULL	2008-08-13

Manager

- Request salary changes for subordinates

Before:

```
SELECT BASE_SALARY FROM EMPLOYEE WHERE LDAP = '3'
```

The screenshot shows a database interface with two panes. The top pane is a 'Query' window containing the following SQL code:

```
1 SELECT BASE_SALARY  
2 FROM EMPLOYEE  
3 WHERE LDAP = '3'  
4
```

The bottom pane is a 'Grid view' showing the results of the query. The table has one column: BASE_SALARY. The data is as follows:

BASE_SALARY
\$73,863.00

After:

```

UPDATE EMPLOYEE SET BASE_SALARY = '100,000.00'
WHERE LDAP = '3'

SELECT BASE_SALARY FROM EMPLOYEE WHERE LDAP = '3'

```

The screenshot shows a database interface with a code editor at the top containing the following SQL code:

```

1 UPDATE EMPLOYEE
2 SET BASE_SALARY = '$100,000.00'
3 WHERE LDAP = '3'
4
5 SELECT BASE_SALARY
6 FROM EMPLOYEE
7 WHERE LDAP = '3'

```

Below the code editor is a toolbar with various icons. The results pane shows the output of the SELECT query:

BASE_SALARY
\$100,000.00

Total rows loaded: 1

- Request promotion/demotion for subordinates

Before:

```
SELECT ROLE_ID FROM EMPLOYEE WHERE LDAP = '4'
```

The screenshot shows a database interface with a code editor at the top containing the following SQL code:

```

1 SELECT ROLE_ID
2 FROM EMPLOYEE
3 WHERE LDAP = '4'

```

Below the code editor is a toolbar with various icons. The results pane shows the output of the SELECT query:

ROLE_ID
R3

Total rows loaded: 1

After:

```
UPDATE EMPLOYEE SET ROLE_ID = 'R2' WHERE LDAP = '4'
```

```
SELECT ROLE_ID FROM EMPLOYEE WHERE LDAP = '4'
```

The screenshot shows a SQL editor window with the following content:

```
1 UPDATE EMPLOYEE
2 SET ROLE_ID = 'R2' WHERE LDAP = '4'
3
4 SELECT ROLE_ID
5 FROM EMPLOYEE
6 WHERE LDAP = '4'
```

Below the editor, a message bar indicates "Total rows loaded: 1". A grid view table is displayed with one row:

ROLE_ID
1 R2

Before:

```
SELECT * FROM ROLE
```

The screenshot shows a SQL editor window with the following content:

```
1 SELECT *
2 FROM ROLE
```

Below the editor, a message bar indicates "Total rows loaded: 7". A grid view table is displayed with the following data:

ROLE_ID	ROLE_TITLE	ROLE_DESCRIPTION
1 R1	VICE PRESIDENT	Oversee and manage all departments and responsible for everything inside the department
2 R2	DEPARTMENT MANAGER	Manager in a department, subordinate might have employee, HR generalist, Manager, HR analy...
3 R3	PROJECT MANAGER	Responsible for managing the project workflow per project requirement. Need to balance const...
4 R4	HR GENERALIST	Oversee company human resources, tasks such as requesting, checking employee's info
5 R5	HR ANALYST	Responsible for analyzing company HR data, and create dash board and insights from those da...
6 R6	APPLICATION ADMIN	Oversee the whole HR application, responsible to checking data of the application, makes chan...
7 R7	EMPLOYEE	Employee of the company, work under employee

After:

```
INSERT INTO ROLE ('ROLE_ID', 'ROLE_TITLE', 'ROLE_DESCRIPTION')
```

```
VALUES ('R8', 'ACCOUNTANT', 'To manage company or department finance related tasks')
```

```
SELECT * FROM ROLE
```

```
1 INSERT INTO ROLE ('ROLE_ID', 'ROLE_TITLE', 'ROLE_DESCRIPTION')
2 VALUES ('R8', 'ACCOUNTANT', 'To manage company or department finance related tasks')
3
4 SELECT *
5 FROM ROLE
```

ROLE_ID	ROLE_TITLE	ROLE_DESCRIPTION
1 R1	VICE PRESIDENT	Oversee and manage all departments and responsible for everything inside the department
2 R2	DEPARTMENT MANAGER	Manager in a department, subordinate might have employee, HR generalist, Manager, HR analy...
3 R3	PROJECT MANAGER	Responsible for managing the project workflow per project requirement. Need to balance const...
4 R4	HR GENERALIST	Oversee company human resources, tasks such as requesting, checking employee's info
5 R5	HR ANALYST	Responsible for analyzing company HR data, and create dash board and insights from those da...
6 R6	APPLICATION ADMIN	Oversee the whole HR application, responsible to checking data of the application, makes chan...
7 R7	EMPLOYEE	Employee of the company, work under employee
8 R8	ACCOUNTANT	To manage company or department finance related tasks

Application Administrator

- Give/change/revoke specific roles for related access

Before:

```
SELECT ROLE_ID FROM EMPLOYEE WHERE LDAP = '7'
```

```
1 SELECT ROLE_ID
2 FROM EMPLOYEE
3 WHERE LDAP = '7'
4
```

ROLE_ID
1 R3

After:

```
UPDATE EMPLOYEE SET ROLE_ID = 'R2' WHERE LDAP = '7'
```

```
SELECT ROLE_ID FROM EMPLOYEE WHERE LDAP = '7'
```

The screenshot shows a database interface with a code editor at the top containing the following SQL statements:

```
1 UPDATE EMPLOYEE
2 SET ROLE_ID = 'R2'
3 WHERE LDAP = '7'
4
5 SELECT ROLE_ID
6 FROM EMPLOYEE
7 WHERE LDAP = '7'
8 |
```

Below the code editor is a toolbar with various icons. The status bar indicates "Total rows loaded: 1". The main area displays a grid view with one row of data:

ROLE_ID
1 R2

- Upon request, will create new entry for projects and departments

Before:

```
SELECT * FROM DEPARTMENT
```

The screenshot shows a database interface with a code editor at the top containing the following SQL statement:

```
1 SELECT *
2 FROM DEPARTMENT
3 |
```

Below the code editor is a toolbar with various icons. The status bar indicates "Total rows loaded: 3". The main area displays a grid view with three rows of data:

DEPT_ID	DEPT_NAME	DEPT_DESCRIPTION	LOCATION_ID	DEPT_MANAGER_ID
1 D1	Organization	Headquarters	L1	1
2 D2	Finance	Records and reports the cash flow transactions of the company	L2	2
3 D3	Technology	Investigate activities a business conducts to improve existing products and procedure	L3	3

After:

```
INSERT INTO DEPARTMENT (DEPT_ID, DEPT_NAME, DEPT_DESCRIPTION,  
LOCATION_ID, DEPT_MANAGER_ID) VALUES ('D4', 'Engineering', 'Design prototype for  
new products', 'L3', 4)
```

```
SELECT * FROM DEPARTMENT
```

The screenshot shows a database interface with a code editor at the top containing the following SQL code:

```
1 INSERT INTO DEPARTMENT (DEPT_ID, DEPT_NAME, DEPT_DESCRIPTION, LOCATION_ID, DEPT_MANAGER_ID)  
2 VALUES ('D4', 'Engineering', 'Design prototype for new products', 'L3', 4);  
3  
4 SELECT *  
5 FROM DEPARTMENT
```

Below the code editor is a table view showing the department data. The table has columns: DEPT_ID, DEPT_NAME, DEPT_DESCRIPTION, LOCATION_ID, and DEPT_MANAGER_ID. The data is as follows:

DEPT_ID	DEPT_NAME	DEPT_DESCRIPTION	LOCATION_ID	DEPT_MANAGER_ID
1 D1	Organization	Headquarters	L1	1
2 D2	Finance	Records and reports the cash flow transactions of the company	L2	2
3 D3	Technology	Investigate activities a business conducts to improve existing products and procedure	L3	3
4 D4	Engineering	Design prototype for new products	L3	4

- Provide audit compliance reports for HR Generalist

```
SELECT FIRST_NAME, LAST_NAME FROM EMPLOYEE
```

```
WHERE ROLE_ID IN (SELECT ROLE_ID FROM ROLE WHERE ROLE_TITLE = 'HR  
GENERALIST')
```

```

94 ---View audit reports by Application Administrator
95 SELECT FIRST_NAME, LAST_NAME
96 FROM EMPLOYEE
97 WHERE ROLE_ID IN (SELECT ROLE_ID FROM ROLE WHERE ROLE_TITLE = 'HR GENERALIST');
98
99
100
101
102
103
104
105
106
107
108
109
110
111

```

Grid view Form view

	FIRST_NAME	LAST_NAME
1	DIANA	ROSS
2	MARY	STEVENS

Total rows loaded: 2

HR Analyst – Queries Discussed in the Business Metrics section below

- Generate different types of business metrics

BUSINESS METRIC QUERIES RESULTS AND REPORTS

Headcount

Assumptions: Only Regular and Contingent employees except inactive employees.

```

SELECT COUNT(LDAP) AS 'TOTAL HEADCOUNT'
FROM EMPLOYEE E
WHERE E.EMPLOYEE_STATUS <> 'INACTIVE' AND (E.EMPLOYEE_TYPE =
'REGULAR' OR E.EMPLOYEE_TYPE = 'CONTINGENT')

```

```

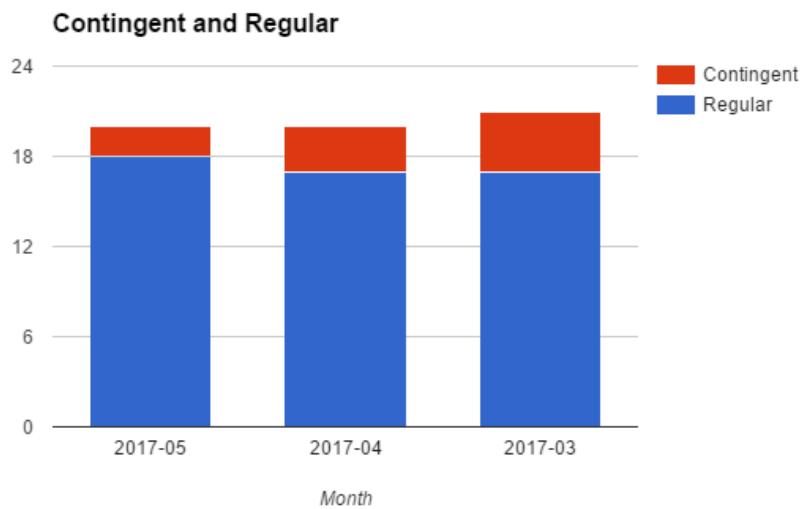
1 SELECT COUNT(LDAP) AS 'TOTAL HEADCOUNT'
2 FROM EMPLOYEE E
3 WHERE E.EMPLOYEE_STATUS <> 'INACTIVE' AND (E.EMPLOYEE_TYPE = 'REGULAR' OR E.EMPLOYEE_TYPE = 'CONTINGENT')

```

Grid view Form view

	TOTAL HEADCOUNT
1	19

Total rows loaded: 1



Span of Control

```

SELECT E.FIRST_NAME,E.LAST_NAME, COUNT(*) AS '# OF EMPLOYEES'
FROM PROJECT_ASSIGNMENT PA, PROJECT_DETAIL PD, DEPARTMENT D,
EMPLOYEE E
WHERE PA.PROJECT_ID=PD.PROJECT_ID AND PD.DEPT_ID=D.DEPT_ID AND
(PROJECT_MANAGER_ID = E.LDAP OR DEPT_MANAGER_ID=E.LDAP)
GROUP BY DEPT_MANAGER_ID,PROJECT_MANAGER_ID

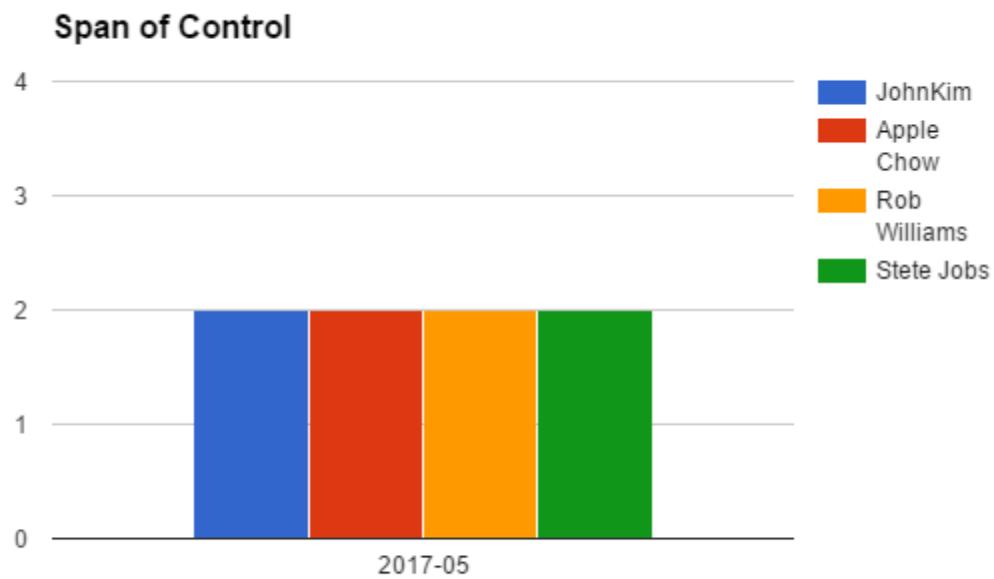
```

```

1 SELECT E.FIRST_NAME,E.LAST_NAME, COUNT(*) AS '# OF EMPLOYEES'
2 FROM PROJECT_ASSIGNMENT PA, PROJECT_DETAIL PD, DEPARTMENT D, EMPLOYEE E
3 WHERE PA.PROJECT_ID=PD.PROJECT_ID AND PD.DEPT_ID=D.DEPT_ID AND (PROJECT_MANAGER_ID = E.LDAP OR DEPT_MANAGER_ID=E.LDAP)
4 GROUP BY DEPT_MANAGER_ID,PROJECT_MANAGER_ID

```

FIRST_NAME	LAST_NAME	# OF EMPLOYEES
1 KIM	JOHN	2
2 APPLE	CHOW	2
3 ROB	WILLIAMS	2
4 STEVE	JOBSS	2



Labor Cost Trend

```

SELECT COUNT(LDAP) AS 'TOTAL_HEADCOUNT' , SUM(BASE_SALARY) AS
'SUM_BASE_SALARY',
AVG(BASE_SALARY) AS 'AVERAGE_BASE_SALARY'
FROM EMPLOYEE E
WHERE E.EMPLOYEE_STATUS <> 'INACTIVE' AND (E.EMPLOYEE_TYPE =
'REGULAR' OR E.EMPLOYEE_TYPE = 'CONTINGENT')

```

```

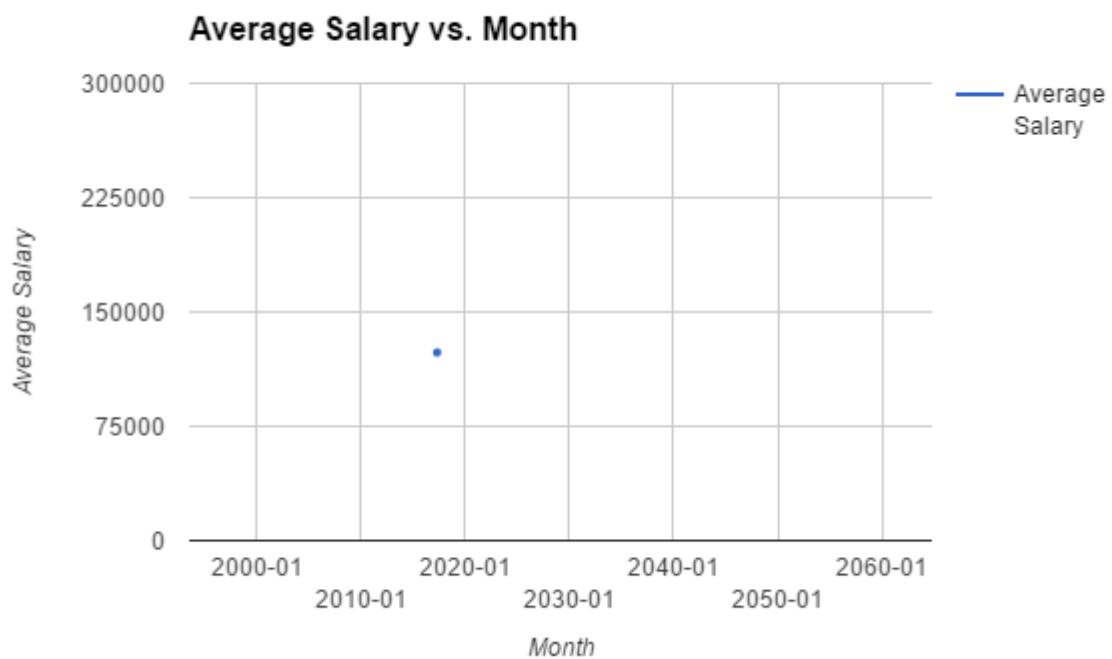
1 SELECT COUNT(LDAP) AS 'TOTAL_HEADCOUNT', SUM(BASE_SALARY) AS 'SUM_BASE_SALARY',
2 AVG(BASE_SALARY) AS 'AVERAGE_BASE_SALARY'
3 FROM EMPLOYEE E
4 WHERE E.EMPLOYEE_STATUS <> 'INACTIVE' AND E.EMPLOYEE_TYPE = 'REGULAR' OR E.EMPLOYEE_TYPE = 'CONTINGENT'

```

Grid view Form view

Total rows loaded: 1

TOTAL_HEADCOUNT	SUM_BASE_SALARY	AVERAGE_BASE_SALARY
18	1929177	107176.5



Termination Count

```

SELECT TOTAL_TERMINATIONS,TOTAL_HEADCOUNT
FROM (
  (SELECT COUNT(LDAP) AS 'TOTAL_HEADCOUNT'

```

```

FROM EMPLOYEE WHERE (EMPLOYEE_STATUS <> 'INACTIVE' AND
DATE_OF_LEAVE LIKE '2017-05%') OR (EMPLOYEE_TYPE = 'REGULAR' OR
EMPLOYEE_TYPE = 'CONTINGENT'))),
(SELECT COUNT(LDAP) AS 'TOTAL_TERMINATIONS' FROM EMPLOYEE WHERE
EMPLOYEE_STATUS = 'TERMINATED' AND DATE_OF_LEAVE LIKE '2017-05%')

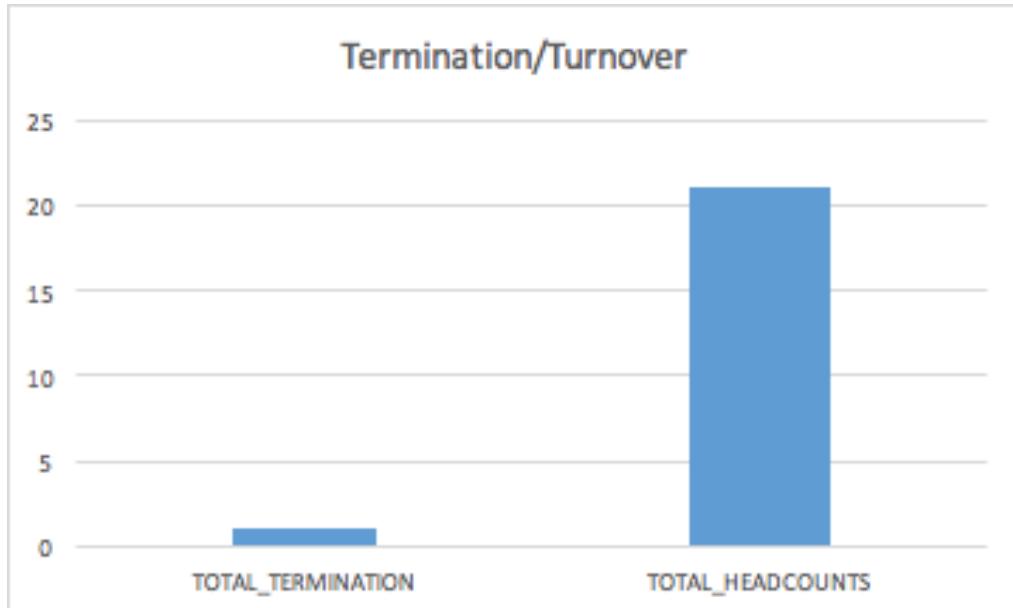
```

```

1 SELECT TOTAL_TERMINATIONS, TOTAL_HEADCOUNT
2 FROM (
3 (SELECT COUNT(LDAP) AS 'TOTAL HEADCOUNT'
4 FROM EMPLOYEE WHERE (EMPLOYEE_STATUS <> 'INACTIVE' AND DATE_OF_LEAVE LIKE '2017-05%') OR (EMPLOYEE_TYPE = 'REGULAR' OR EMPLOYEE_TYPE = 'CONTINGENT'))),
5 (SELECT COUNT(LDAP) AS 'TOTAL_TERMINATIONS' FROM EMPLOYEE WHERE EMPLOYEE_STATUS = 'TERMINATED' AND DATE_OF_LEAVE LIKE '2017-05%')
6

```

		Total rows loaded: 1
	TOTAL_TERMINATIONS	TOTAL_HEADCOUNT
1	1	21



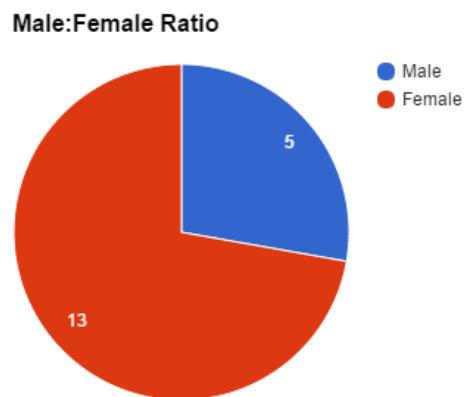
Workforce Gender Metrics

```
SELECT GENDER, COUNT(LDAP) AS 'TOTAL_EMPLOYEE'  
FROM EMPLOYEE E  
WHERE E.EMPLOYEE_STATUS <> 'INACTIVE' AND (E.EMPLOYEE_TYPE =  
'REGULAR' OR E.EMPLOYEE_TYPE = 'CONTINGENT')  
GROUP BY GENDER
```

```
1 SELECT GENDER, COUNT(LDAP) AS 'TOTAL_EMPLOYEE'  
2 FROM EMPLOYEE E  
3 WHERE E.EMPLOYEE_STATUS <> 'INACTIVE' AND (E.EMPLOYEE_TYPE = 'REGULAR' OR E.EMPLOYEE_TYPE = 'CONTINGENT')  
4 GROUP BY GENDER
```

The screenshot shows a database interface with a query editor at the top containing the SQL code. Below it is a results grid with two rows of data. The grid has columns for 'GENDER' and 'TOTAL_EMPLOYEE'. The first row shows 'FEMALE' with a value of 14, and the second row shows 'MALE' with a value of 5. At the bottom of the grid, it says 'Total rows loaded: 2'. Above the grid are standard database navigation icons like back, forward, and search. Below the grid are 'Grid view' and 'Form view' buttons.

GENDER	TOTAL_EMPLOYEE
1 FEMALE	14
2 MALE	5



Number of Hires/Terminations/Promotions/Transfers

```
SELECT COUNT(*) , RD.REQUEST_TITLE
```

```

FROM REQUEST RA
INNER JOIN REQUEST_TABLE RD
ON RA.REQUEST_ID=RD.REQUEST_ID
WHERE RD.REQUEST_ID ='RQ1' OR RD.REQUEST_ID ='RQ2' OR RD.REQUEST_ID
='RQ3' OR RD.REQUEST_ID ='RQ7'
GROUP BY REQUEST_TITLE

```

```

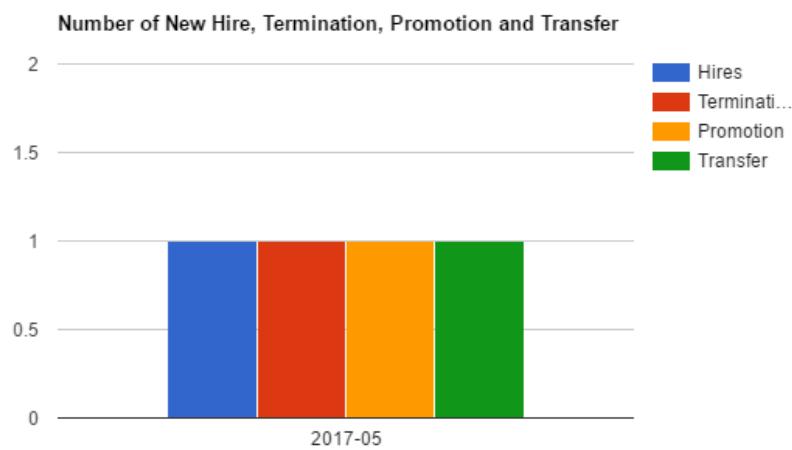
1 SELECT COUNT(*) , RD.REQUEST_TITLE
2 FROM REQUEST RA
3 INNER JOIN REQUEST_TABLE RD
4 ON RA.REQUEST_ID=RD.REQUEST_ID
5 WHERE RD.REQUEST_ID ='RQ1' OR RD.REQUEST_ID ='RQ2' OR RD.REQUEST_ID ='RQ3' OR RD.REQUEST_ID ='RQ7'
6 GROUP BY REQUEST_TITLE
7

```

Grid view Form view

Total rows loaded: 4

COUNT(*)	REQUEST_TITLE
1	HIRE
1	PROMOTION/DEMOTION(EMPLOYEE)
1	TERMINATE
1	TRANSFER



PROJECT SUMMARY

Experience

This project was meaningful and insightful for us in the following ways:

- Application of the theoretical DBMS concepts learnt in the course to a real-life business application.
- Conceptualization of a business scenario and developing the appropriate use cases enhanced our learning curve and taught us how to narrow our focus for the required items thereby beating the temptation of generalizing our scope.
- Designing the different schemas and generating the UML with the correct entities, relationships and cardinality was the most enriching experience.
- Building a data dictionary was a very fruitful exercise as that brought the complexities of the referential integrities and related dependencies that trained us around handling actual data.
- Working in a team was good experience in terms of handling different perspectives that we may often miss when working individually.

Overall, this exercise was a great value addition to our technical knowledge as we learned to build a DBMS model from ground and develop it into a working prototype.

Challenges

The hardest part of the project was building the UML diagram. To get the correct entities and develop relations along with the cardinalities was extremely challenging for an actual business scenario. The second challenge was generation of data. This was complicated as we had to create different scenarios while mapping the referential integrity constraints. Lastly tracking the historical data on a time basis was hard as we did not implement a system date table.

Learnings

Our initial attempt in designing our application was an ambitious attempt. This was difficult to achieve and especially hard in terms of designing the schema, UML. With a few iterations and guidance from Professor Agarwal, we were able to modify and narrow our schema and proceed further to reach a logical point. For creating Indexes and enforcing Referential Integrity, we referred the class presentations and worked together to overcome the related challenges.

If we do this whole project again, we would focus on the UML more than the users and use cases and will make sure we don't think about the tables before we finalize our UML, otherwise we will have to keep changing our tables and data based on UML changes.

Suggestions

Overall, this was a great project. I feel the project is carefully formulated and has the correct milestones. If possible, we would like to work with real life data sets that would be an added learning.

Future Opportunities

For our application, we would like to incorporate system date tables, triggers, views, transactions and historical logs. We would also include other Human Resource functions like Compensation, Recruitment or performance management. This would help us to calculate derived metrics and eventually help the company make informed data driven decisions for the betterment of the company and workforce.