MSIS 2603 / OMIS 3366 Final Project

"HR Wizard"

Jonathan Ang ▪ Shivam Kapoor ▪ Huey Veltchev ▪ Kyle Wright

June 6, 2018

Santa Clara University

Table of Contents

Introduction

Our team's objective is to design a Human Resources Application, 'HR Wizard', that performs Payroll Management and Training Management. The client company that we are developing 'HR Wizard' for is Uno Web Applications, located in Silicon Valley, CA.

Uno Web Applications is founded in 2013 and currently has 26 employees on its staff. There are three departments in Uno, namely Engineering, Information Technology (IT), and Human Resources (HR).

Uno aims to grow its headcounts, albeit conservatively, due to limited financial backing. It was founded with and is currently operating off of its founder and Engineering Manager, Randall Holt's personal savings.

Type of Users

1. Payroll Clerk

The is a role in the Human Resources Department. This person is responsible for paying employees in the company on time. He or she performs calculations and dissemination of salaries.

2. Functional Manager

This is a People Manager role for a function or department (Engineering, IT or HR), also known as the Line Manager for said function or department.

3. Salaried Employee

This is an employee who works in the company, within a specific department i.e. Engineering, IT, or HR, and reports to a Functional Manager.

4. Training Manager

This role is in the Human Resources Department. He or she supports the Functional Manager in employee training management i.e. training planning, schedule classes, report performance on classes.

Use Cases

1. Payroll Clerk

1. Log in to and out of the application.
2. Look up employees that require updates on their payroll account.
3. Add incoming employees, and update status for outgoing employees to ‘Inactive’.
4. Look up employees that require updates on their payroll account.
5. Calculate benefit amount based on employee’s benefit options.
6. Calculate and issue salary based on employee’s pay grade, reimbursement amount, and PTO.
7. Calculate and issue employee’s reimbursements based manager’s approval.

2. Functional Manager

1. Log in to and out of the application.
2. Look up employees that have applied for PTO or sick time in the application.
3. Approve PTO, sick time, and reimbursement requests.
4. Look up employee reimbursement requests.
5. Approve or disapprove employee reimbursements requests.
6. Update employee’s pay grade in case of promotion.

3. Salaried Employee

1. Log in to and out of the application.
2. View pay stubs and pay history.
3. View training options (types of courses, schedule, and location) in the application.
4. View schedule of training sessions registered.
5. Create, submit, and track reimbursement requests.
6. Create, submit, and track PTO requests.

4. Training Manager

1. Log in to and out of the application.
2. Add, update, and look up training courses for new hires or existing employees.
3. Schedule, update, and look up employees’ enrollment in training courses.
4. Report training statistics to functional managers.

Business Metrics

1. Growth Rate

"HR Wizard" provides its client companies with the ability to calculate growth rate by returning the necessary data. See the graph below that shows the trend of Uno's growth for the years 2013 to 2018 using said data.

2. Attrition Rate

"HR Wizard" provides its client companies with the ability to calculate the attrition rate by returning the number of employees leaving the company per year. Graph below shows the trend of Uno's attrition rate for the years 2013 to 2018 using said data.

3. Benefit Enrollment Rate

The percentage of Benefit Enrollment is calculated by:

$$\frac{\#\ of\ employees\ enrolled\ in\ benefit(s)}{\#\ of\ employees} \times 100$$

This metric gives client companies the insight into what they can improve on, in terms of benefits offered, for attracting the talents that they are looking for.

4. Training Participation Rate

The percentage of Training Participation Rate is calculated by:

$$\frac{\#\ employees\ took\ at\ least\ one\ training\ course}{\#\ of\ employees} \times 100$$

This metric gives client companies the insight into what they can improve on, in terms of training, for attracting more employees to participate in courses offered.

Enumeration of Queries

1. Payroll Clerk

1. **Search**. Search the list of employees who are enrolled in benefits. Return the name of the employees and their departments.
2. **Update**. Increase salary by 10% for employees who have worked for over 3 years. Return the name of the employees, their IDs and new salary.
3. **Insert**. Add new employee Kyle Wright. Employee_id = 17, manager_id = 1, city is Santa Clara, street address is 430 El Camino Real, state is CA, zip is 95040, phone number is (909) 590-3429, role is Salaried Employee, pay grade is C, department is Engineering, manager ID is 1, start date is 6-3-2018, end date is NULL and status is Active.

2. Functional Manager

1. **Search.** Search all employees whose time-off approval status is DENIED.
2. **Update.** Change all time-off approval status DENIED to APPROVED.
3. **Search.** Find the number of expense submissions that are denied and the employee name that submitted them.

3. Salaried Employee

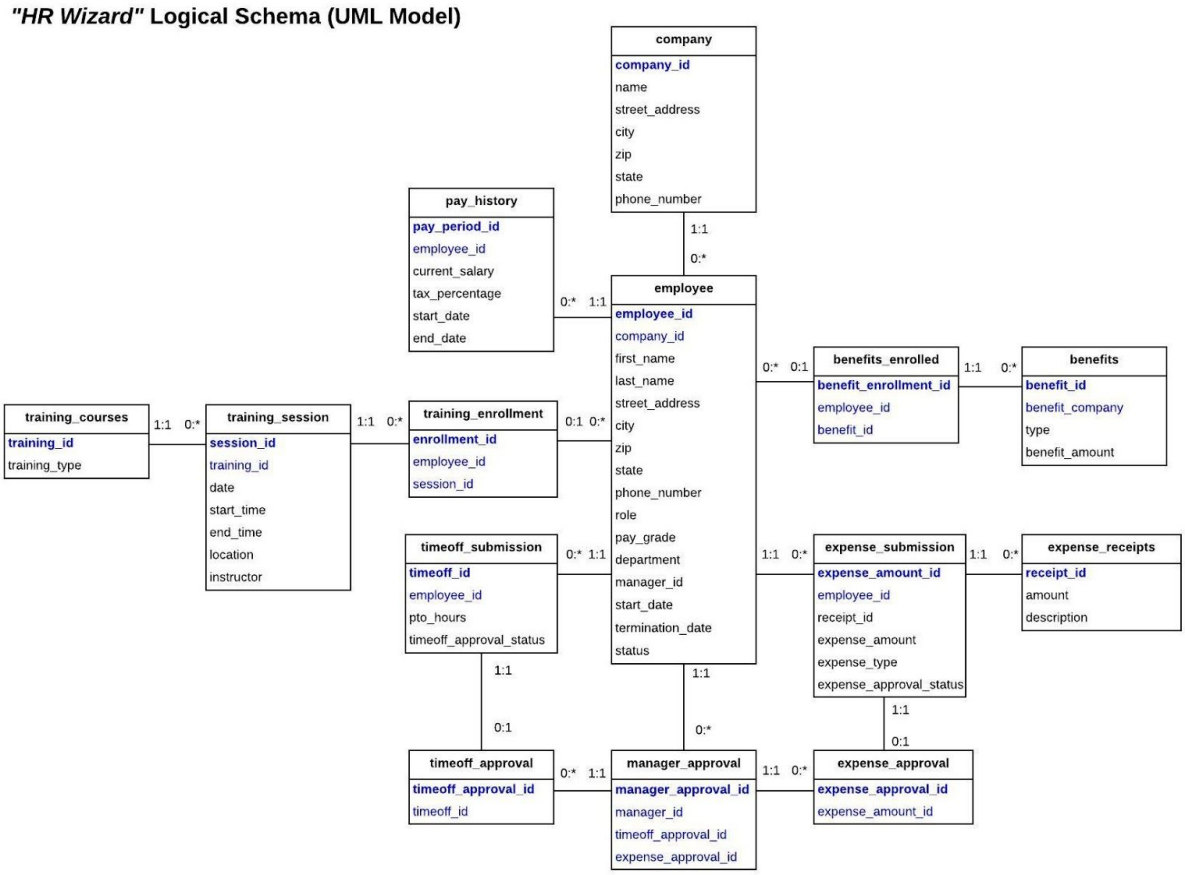
1. **Search.** Return the name of all managers in Engineering.
2. **Search.** Return the name of all managers in Engineering and the list of approved time-off requests for each.
3. **Insert.** Ursula Herring wants to take a 24-hour time off. Insert timeoff_id T7, PTO hours 24, timeoff_approval_status is NULL to employee ID 7.
4. **Update.** Update Randall Holt's phone number to (800) 432-4309.
5. **Delete.** Delete Lee Riley's time-off request.

4. Training Manager

1. **Search.** Return the list of employees who are enrolled in training courses. Return the name of the employees, department, course(s) enrolled, and the date and time of the course(s).
2. **Insert.** Insert training ID T6 and training type 'Finance' into the list of training courses.
3. **Update.** Change training session S1 to be in the Purple room.
4. **Delete.** Delete all attributes in training session where session ID is S5.

Logical Schema

The diagram below shows the logical schema or UML model for “HR Wizard”. All primary keys are in **bold and blue**, and all foreign keys are in blue.



Physical Schema

We present the physical schema, which is also known as the database dictionary, by table, in this section. Most attributes for a table are self-explanatory, additional description is only available for those that are not.

company

Attribute	Type	Constraints	Additional Description
company_ID	VARCHAR(20)	Primary Key	*- attribute is self-explanatory -*
name	VARCHAR(20)	NOT NULL	*- attribute is self-explanatory -*
street_address	VARCHAR(50)	NOT NULL	*- attribute is self-explanatory -*
city	VARCHAR(50)	NOT NULL	*- attribute is self-explanatory -*
zip	VARINT(5)	NOT NULL	*- attribute is self-explanatory -*
state	VARCHAR(30)	NOT NULL	*- attribute is self-explanatory -*
phone_number	VARCHAR(10)	NOT NULL	*- attribute is self-explanatory -*

employee

Attribute	Type	Constraints	Additional Description
employee_ID	VARCHAR(20)	Primary Key	*- attribute is self-explanatory -*
company_ID	VARCHAR(20)	Foreign Key	*- attribute is self-explanatory -*
first_name	VARCHAR(20)	NOT NULL	*- attribute is self-explanatory -*
last_name	VARCHAR(20)	NOT NULL	*- attribute is self-explanatory -*
street_address	VARCHAR(50)	NOT NULL	*- attribute is self-explanatory -*
city	VARCHAR(50)	NOT NULL	*- attribute is self-explanatory -*
zip	VARINT(5)	NOT NULL	*- attribute is self-explanatory -*
state	VARCHAR(30)	NOT NULL	*- attribute is self-explanatory -*
phone_number	VARCHAR(10)	NOT NULL	*- attribute is self-explanatory -*
role	VARCHAR(50)	NOT NULL	*- attribute is self-explanatory -*
pay_grade	FLOAT	NOT NULL	*- attribute is self-explanatory -*
department	VARCHAR(50)	NOT NULL	*- attribute is self-explanatory -*
manager_ID	VARCHAR(20)	NOT NULL	*- attribute is self-explanatory -*
start_date	DATE	NOT NULL	*- attribute is self-explanatory -*
termination_date	DATE	NOT NULL	*- attribute is self-explanatory -*
status	VARCHAR(20)	NOT NULL	Active' for current employees; 'Inactive' for former employees

pay_history

Attribute	Type	Constraints	Additional Description
pay_period_ID	VARCHAR(20)	Primary Key	Pay period for a paycheck
employee_ID	VARCHAR(20)	Foreign Key	- *attribute is self-explanatory* -
current_salary	FLOAT	NOT NULL	- *attribute is self-explanatory* -
tax_percentage	FLOAT	NOT NULL	- *attribute is self-explanatory* -
start_date	DATE	NOT NULL	Start date of pay period
end_date	DATE	NOT NULL	End date of pay period

training_enrollment

Attribute	Type	Constraints	Additional Description
enrollment_ID	VARCHAR(20)	Primary Key	- *attribute is self-explanatory* -
employee_ID	VARCHAR(20)	Foreign Key	- *attribute is self-explanatory* -
session_ID	VARCHAR(20)	Foreign Key	ID for a session an employee is enrolled in

training_session

Attribute	Type	Constraints	Additional Description
session_id	VARCHAR(20)	Primary Key	There can only be one course offered during a session (at a date and time)
training_id	VARINT(20)	Foreign Key	ID for the training course that an employee is enrolled in
date	DATE	NOT NULL	- *attribute is self-explanatory* -
start_time	TIME	NOT NULL	Start time for the training session
end_time	TIME	NOT NULL	End time for the training session
location	VARCHAR(30)	NOT NULL	Location for the training
instructor	VARCHAR(50)	NOT NULL	- *attribute is self-explanatory* -

training_courses

Attribute	Type	Constraints	Additional Description
training_ID	VARINT(20)	Primary Key	- *attribute is self-explanatory* -
training_type	VARCHAR(30)	Foreign Key	The type of training i.e. Engineering, Human Resource, Payroll, IT, or Expenses

benefit_enrolled

Attribute	Type	Constraints	Additional Description

benefit_ID	VARINT(20)	Primary Key	*- attribute is self-explanatory -*
employee_ID	VARCHAR(20)	Foreign Key	*- attribute is self-explanatory -*

benefits

Attribute	Type	Constraints	Additional Description
benefit_ID	VARCHAR(20)	Primary Key	*- attribute is self-explanatory -*
benefit_company	VARCHAR(50)	NOT NULL	*- attribute is self-explanatory -*
benefit_type	VARCHAR(30)	NOT NULL	The types of benefits available i.e. 401K, car, health, life, or disability
benefit_amount	FLOAT	NOT NULL	*- attribute is self-explanatory -*

expenese_submission

Attribute	Type	Constraints	Additional Description
expense_amount_ID	VARCHAR(20)	Primary Key	*- attribute is self-explanatory -*
employee_ID	VARCHAR(20)	Foreign Key	*- attribute is self-explanatory -*
receipt_ID	VARCHAR(20)	Foreign Key	*- attribute is self-explanatory -*
expense_amount	FLOAT	NOT NULL	*- attribute is self-explanatory -*
expense_type	VARCHAR(20)	NOT NULL	The type of expense i.e. cash or credit
expense_approval_status	VARCHAR(20)	NOT NULL	The outcome of the approval process is either 'Approved' or 'Denied'

expense_receipt

Attribute	Type	Constraints	Additional Description
receipt_ID	VARCHAR(20)	Primary Key	*- attribute is self-explanatory -*
amount	FLOAT	NOT NULL	*- attribute is self-explanatory -*
description	VARCHAR(140)	NULL	The description of the expense i.e. food, travel, etc.

expense_approval

Attribute	Type	Constraints	Description
expense_approval_ID	VARCHAR(20)	Primary Key	*- attribute is self-explanatory -*
expense_amount_ID	VARCHAR(20)	Foreign Key	*- attribute is self-explanatory -*

timeoff_submission

Attribute	Type	Constraints	Additional Description

timeoff_ID	VARCHAR(20)	Primary Key	*- attribute is self-explanatory -*
employee_ID	VARCHAR(20)	Foreign Key	*- attribute is self-explanatory -*
pto_hours	FLOAT	NOT NULL	The number of PTO hours requested by an employee
timeoff_approval_status	VARCHAR(20)	NOT NULL	The outcome of the approval process is either 'Approved' or 'Denied'

timeoff_approval

Attribute	Type	Constraints	Additional Description
timeoff_approval_ID	VARCHAR(20)	Primary Key	*- attribute is self-explanatory -*
timeoff_ID	VARCHAR(20)	Foreign Key	*- attribute is self-explanatory -*

manager_approval

Attribute	Type	Constraints	Additional Description
manager_approval_ID	VARCHAR(20)	Primary Key	The serial number generated to store all the approval submitted for approvals
manager_ID	VARCHAR(20)	Foreign Key	Links the employee table for manager and manager's approval
timeoff_approval_ID	VARCHAR(20)	Foreign Key	Links with the expense_approval table to approve the submitted expenses in the timeoff_submission table
expense_approval_ID	VARCHAR(20)	Foreign Key	Links the expense_approval table with the expense_submission table, to approve the submitted expenses from the expense_submission table

SQL Queries by Role

1. Payroll Clerk

1. **Search**. Search the list of employees who are enrolled in benefits. Return name of employees and departments.

```
1  SELECT e.employee_id AS 'Employee ID', e.first_name AS 'First Name',
2  e.last_name AS 'Last Name', e.department AS 'Department'
3  FROM employee e
4  WHERE EXISTS(SELECT be.benefit_id
5               FROM benefits_enrolled be, employee e
6               WHERE e.employee_id = be.employee_id
7               ORDER BY e.employee_id DESC)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Employee ID	First Name	Last Name	Department
1	Randall	Holt	Engineering
10	Emerald	Velazquez	Information Technology
11	Alexis	Wolf	Engineering
12	Rhonda	Mckee	Information Technology
13	Hope	Ashley	Information Technology
14	Nola	Kirk	Human Resources
15	Karen	Kent	Information Technology
16	Sybil	Chandler	Information Technology
2	Cedric	Dotson	Engineering
3	Ethan	Pruitt	Engineering
4	Conan	Greer	Human Resources
5	Ralph	Hurley	Information Technology
6	Camilla	Sutton	Human Resources
7	Ursula	Herring	Engineering
8	Jescie	Logan	Engineering
9	Lee	Riley	Information Technology

2. **Update**. Increase salary by 10% for employees who have worked for over 3 years.Return the name of employees, their IDs and new salary.

   Before Update:

```
UPDATE pay_history p, employee e
SET p.current_salary = p.current_salary * 1.1
WHERE e.employee_id = p.employee_id and
e.start_date < '2015/06/01'
```

First Name	Last Name	Start Date	Salary
Randall	Holt	2013-02-16	4657.95
Ethan	Pruitt	2014-05-29	3108.6
Conan	Greer	2014-04-03	3861
Ralph	Hurlev	2013-10-16	4657.95
Camilla	Sutton	2013-12-01	2326.5
Jescie	Logan	2013-07-11	3108.6
Rhonda	Mckee	2014-06-01	2722.5
Karen	Kent	2013-05-30	2722.5

After Update:

```
1  SELECT DISTINCT e.FIRST_NAME AS 'First Name', e.LAST_NAME AS 'Last Name',
2  e.start_date AS 'Start Date', p.current_salary as 'Salary'
3  FROM employee e, pay_history p
4  WHERE e.employee_id = p.employee_id AND
5  e.start_date < '2015/06/01'
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

First Name	Last Name	Start Date	Salary
Randall	Holt	2013-02-16	5123.75
Ethan	Pruitt	2014-05-29	3419.46
Conan	Greer	2014-04-03	4247.1
Ralph	Hurlev	2013-10-16	5123.75
Camilla	Sutton	2013-12-01	2559.15
Jescie	Logan	2013-07-11	3419.46
Rhonda	Mckee	2014-06-01	2994.75
Karen	Kent	2013-05-30	2994.75

3. **Insert**. Add new employee Kyle Wright. Employee_id = 17, manager_id = 1, city is Santa Clara, street address is 430 El Camino Real, state is CA, zip is 95040, phone number is (909) 590-3429, role is Salaried Employee, pay grade is C, department is Engineering, manager ID is 1, start date is 6-3-2018, end date is NULL and status is Active.

Before Insert:

EMPLOYEE_ID	COMPANY_ID	FIRST_NAME	LAST_NAME	STREET_ADDRESS	CITY	ZIP
1	1	Randall	Holt	4371 Lobortis Avenue	Fresno	90430
10	1	Emerald	Velazquez	P.O. Box 934, 7718 A. Avenue	Pittsburgh	52968
11	1	Alexis	Wolf	P.O. Box 871, 7906 Laoreet Rd.	Harrisburg	92809
12	1	Rhonda	Mckee	583-3852 Tellus Rd.	Allentown	89621
13	1	Hope	Ashley	Ap #492-8640 Ornare, Rd.	Tucson	85184
14	1	Nola	Kirk	5946 Faucibus Ave	San Jose	94253
15	1	Karen	Kent	3302 Maecenas St.	Pittsburgh	47394
16	1	Sybil	Chandler	P.O. Box 231, 9794 Pharetra, Avenue	Mesa	85034
2	1	Cedric	Dotson	6079 Aliquam Street	Allenton	62624
3	1	Ethan	Pruitt	384-388 Commodo St.	Phoenix	86695
4	1	Conan	Greer	Ap #417-8677 Tempus Rd.	Sacramento	95339
5	1	Ralph	Hurley	351-2501 Tortor, Avenue	Sacramento	95339
6	1	Camilla	Sutton	P.O. Box 796, 6522 Tempus Road	Sacramento	96136
7	1	Ursula	Herring	828-8490 Lorem, Avenue	San Jose	94495
8	1	Jescie	Logan	P.O. Box 417, 7843 Elementum Rd.	San Diego	92809
9	1	Lee	Riley	Ap #804-4748 Odio Avenue	Chandler	85034
NULL	NULL	NULL	NULL	NULL	NULL	NULL

After Insert:

```
INSERT INTO `hr_wizard`.`employee`
(`employee_id`, `company_id`,
`first_name`, `last_name`,
`STREET_ADDRESS`,
`city`,
`zip`,
`state`,
`Phone_number`, `role`, `department`,
`manager_id`, `status`, `pay_grade`, `start_date`, `end_date`)
VALUES
("17", "1", "Kyle", "Wright", "430 El Cmaino Real",
"Santa Clara", 95040, "CA", "9095903429", "Salaried Employee", "Engineering",
1, "ACTIVE", "C", "2018/6/3", NULL);
```

EMPLOYEE_ID	COMPANY_ID	FIRST_NAME	LAST_NAME	STREET_ADDRESS	CITY	ZIP
1	1	Randall	Holt	4371 Lobortis Avenue	Fresno	90430
10	1	Emerald	Velazquez	P.O. Box 934, 7718 A. Avenue	Pittsburgh	52968
11	1	Alexis	Wolf	P.O. Box 871, 7906 Laoreet Rd.	Harrisburg	92809
12	1	Rhonda	Mckee	583-3852 Tellus Rd.	Allentown	89621
13	1	Hope	Ashley	Ap #492-8640 Ornare, Rd.	Tucson	85184
14	1	Nola	Kirk	5946 Faucibus Ave	San Jose	94253
15	1	Karen	Kent	3302 Maecenas St.	Pittsburgh	47394
16	1	Sybil	Chandler	P.O. Box 231, 9794 Pharetra, Avenue	Mesa	85034
17	1	Kyle	Wright	430 El Cmaino Real	Santa Clara	95040
2	1	Cedric	Dotson	6079 Aliquam Street	Allenton	62624
3	1	Ethan	Pruitt	384-388 Commodo St.	Phoenix	86695
4	1	Conan	Greer	Ap #417-8677 Tempus Rd.	Sacramento	95339
5	1	Ralph	Hurley	351-2501 Tortor, Avenue	Sacramento	95339
6	1	Camilla	Sutton	P.O. Box 796, 6522 Tempus Road	Sacramento	96136
7	1	Ursula	Herring	828-8490 Lorem, Avenue	San Jose	94495
8	1	Jescie	Logan	P.O. Box 417, 7843 Elementum Rd.	San Diego	92809
9	1	Lee	Riley	Ap #804-4748 Odio Avenue	Chandler	85034
NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Functional Manager

1. **Search.** Search all employees whose time-off approval status is DENIED.

```
SELECT e.employee_id as 'Employee ID', e.first_name as 'First Name', e.last_name
as 'Last Name', t.timeoff_id as 'Timeoff ID', t.timeoff_approval_status as 'Status'
FROM employee e, timeoff_submission t
WHERE t.timeoff_approval_status = "DENIED" and
e.employee_id = t.employee_id
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

Employee ID	First Name	Last Name	Timeoff ID	Status
9	Lee	Rilev	T4	DENIED
14	Nola	Kirk	T6	DENIED

2. **Update.** Change all time-off approval status DENIED to APPROVED.

Before Update:

timeoff_id	employee_id	pto_hours	timeoff_approval_status
T1	2	20	APPROVED
T2	3	4	APPROVED
T3	8	15	APPROVED
T4	9	53	DENIED
T5	12	15	APPROVED
T6	14	8	DENIED
NULL	NULL	NULL	NULL

After Update:

```
UPDATE timeoff_submission t
SET t.timeoff_approval_status = "APPROVED"
WHERE t.timeoff_approval_status = "DENIED"
```

timeoff_id	employee_id	pto_hours	timeoff_approval_status
T1	2	20	APPROVED
T2	3	4	APPROVED
T3	8	15	APPROVED
T4	9	53	APPROVED
T5	12	15	APPROVED
T6	14	8	APPROVED
NULL	NULL	NULL	NULL

3. **Search.** Find the number of expense submissions that are denied and the employee name that submitted them.

```
SELECT e.employee_id as 'employee ID', e.first_name as 'First Name', e.last_name as 'Last Name',
count(es.expense_amount_id) as 'Number Denied'
FROM employee e, expense_submission es
WHERE e.employee_id = es.employee_id and es.expense_approval_status = "DENIED"
GROUP BY e.employee_id
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

employee ID	First Name	Last Name	Number Denied
15	Karen	Kent	1
9	Lee	Riley	1

3. Salaried Employee

1. **Search.** Return the name of all managers in Engineering.

```
SELECT e.first_name as 'First Name', e.last_name as 'Last Name',
e.department as 'Department', e.manager_id as 'Manager ID'
FROM employee e
WHERE e.manager_id = 0 and e.department = "ENGINEERING"
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

First Name	Last Name	Department	Manager ID
Randall	Holt	Engineering	0

2. **Search.** Return the name of all managers in Engineering and the list of approved time-off requests for each.

```
SELECT e.first_name as 'First Name', e.last_name as 'Last Name',
e.department as 'Department', e.manager_id as 'Manager ID',
m.timeoff_approval_id as 'Time off Approval ID', t.timeoff_approval_status as
'Time off Status'
FROM employee e, manager_approval m, timeoff_submission t, timeoff_approval ta
WHERE e.department = "ENGINEERING" and e.manager_id = m.manager_id and
t.timeoff_approval_status = "APPROVED" and ta.timeoff_id = t.timeoff_id
GROUP BY m.manager_approval_id
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

First Name	Last Name	Department	Manager ID	Time off Approval ID	Time off Status
Randall	Holt	Engineering	1	TO1	APPROVED
Randall	Holt	Engineering	1	TO2	APPROVED
Randall	Holt	Engineering	1	TO3	APPROVED

3. **Insert**. Ursula Herring wants to take a 24-hour time off. Insert timeoff_id T7, PTO hours 24, timeoff_approval_status is NULL to employee ID 7.

Before Insert:

timeoff_id	employee_id	pto_hours	timeoff_approval_status
T1	2	20	APPROVED
T2	3	4	APPROVED
T3	8	15	APPROVED
T4	9	53	APPROVED
T5	12	15	APPROVED
T6	14	8	APPROVED
NULL	NULL	NULL	NULL

After Insert:

```
INSERT INTO `hr_wizard`.`timeoff_submission`
(`timeoff_id`,
`employee_id`,
`pto_hours`,
`timeoff_approval_status`)
VALUES
("T7",
7,
24,
NULL);
```

timeoff_id	employee_id	pto_hours	timeoff_approval_status
T1	2	20	APPROVED
T2	3	4	APPROVED
T3	8	15	APPROVED
T4	9	53	APPROVED
T5	12	15	APPROVED
T6	14	8	APPROVED
T7	7	24	NULL
NULL	NULL	NULL	NULL

4. **Update**. Update Randall Holt's phone number to (800) 432-4309.

Before Update:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
1	Randall	Holt	8323846394
10	Emerald	Velazquez	5026170915
11	Alexis	Wolf	8328800730
12	Rhonda	Mckee	4317251508
13	Hope	Ashley	8749137625
14	Nola	Kirk	9137267242
15	Karen	Kent	4795662637
16	Sybil	Chandler	6728290344
17	Kyle	Wright	9095903429
2	Cedric	Dotson	8636477214
3	Ethan	Pruitt	5975097158
4	Conan	Greer	9818751257
5	Ralph	Hurley	6863120469
6	Camilla	Sutton	9071649893
7	Ursula	Herring	5055887372
8	Jessie	Logan	7735565593
9	Lee	Riley	1497209003

After Update:

```
UPDATE employee
SET phone_number = "8004324309"
WHERE first_name = "Randall" and last_name = "Holt"
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
1	Randall	Holt	8004324309
10	Emerald	Velazquez	5026170915
11	Alexis	Wolf	8328800730
12	Rhonda	Mckee	4317251508
13	Hope	Ashley	8749137625
14	Nola	Kirk	9137267242
15	Karen	Kent	4795662637
16	Sybil	Chandler	6728290344
17	Kyle	Wright	9095903429
2	Cedric	Dotson	8636477214
3	Ethan	Pruitt	5975097158
4	Conan	Greer	9818751257
5	Ralph	Hurley	6863120469
6	Camilla	Sutton	9071649893
7	Ursula	Herring	5055887372
8	Jessie	Logan	7735565593
9	Lee	Riley	1497209003

5. **Delete**. Delete Lee Riley's time-off request.

Before Deletion:

timeoff_id	employee_id	pto_hours	timeoff_app
T1	2	20	APPROVED
T2	3	4	APPROVED
T3	8	15	APPROVED
T4	9	53	APPROVED
T5	12	15	APPROVED
T6	14	8	APPROVED
T7	7	24	NULL
NULL	NULL	NULL	NULL

After Deletion:

```
DELETE FROM timeoff_submission
WHERE employee_id = 9
```

timeoff_id	employee_id	pto_hours	timeoff_approval_status
T1	2	20	APPROVED
T2	3	4	APPROVED
T3	8	15	APPROVED
T5	12	15	APPROVED
T6	14	8	APPROVED
T7	7	24	NULL
NULL	NULL	NULL	NULL

4. Training Manager

1. **Search**. Return the list of employees who are enrolled in training courses. Return the name of the employees, department, course(s) enrolled, and the date and time of the course(s).

```
SELECT e.first_name as 'First Name', e.last_name as 'Last Name',
e.department as 'Department', tc.training_type as 'Course Description', ts.date 'Date',
ts.start_time 'Start Time'
FROM employee e, training_enrollment te, training_session ts, training_courses tc
WHERE e.employee_id = te.employee_id and te.session_id = ts.session_id and
ts.training_id = tc.training_id
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

First Name	Last Name	Department	Course Description	Date	Start Time
Cedric	Dotson	Engineering	Engineering	2015-07-14	08:00:00
Hope	Ashley	Information Technology	INFORMATION TECHNOLOGY	2017-06-22	08:00:00
Ethan	Pruitt	Engineering	Engineering	2014-05-29	08:00:00
Camilla	Sutton	Human Resources	Human Resource	2013-12-01	08:00:00
Ursula	Herring	Engineering	Expense	2017-08-31	08:00:00
Jescie	Logan	Engineering	Engineering	2013-07-11	08:00:00
Lee	Riley	Information Technology	INFORMATION TECHNOLOGY	2018-05-01	08:00:00
Emerald	Velazquez	Information Technology	Pay Roll	2015-06-28	08:00:00
Alexis	Wolf	Engineering	Engineering	2016-05-24	08:00:00
Rhonda	Mckee	Information Technology	Pay Roll	2014-06-01	08:00:00

2. **Insert**. Insert training ID T6 and training type ‘Finance’ into the list of training courses.

Before Insert:

training_id	training_type
T1	Engineering
T2	Human Resource
T3	Pay Roll
T4	INFORMATION TECHNOLOGY
T5	Expense
NULL	NULL

After Insert:

```
INSERT INTO training_courses
(`training_id`,
`training_type`)
VALUES
("T6",
"FINANCE");
```

training_id	training_type
T1	Engineering
T2	Human Resource
T3	Pay Roll
T4	INFORMATION TECHNOLOGY
T5	Expense
T6	FINANCE
NULL	NULL

3. **Update.** Change training session S1 to be in the Purple room.

Before Update:

session_id	training_id	date	start_time	end_time	location	instructor
S1	T1	2015-07-14	08:00:00	13:00:00	Red Room	Randall
S10	T4	2017-06-22	08:00:00	13:00:00	Blue Room	Ralph
S2	T1	2014-05-29	08:00:00	13:00:00	Red Room	Randall
S3	T2	2013-12-01	08:00:00	13:00:00	Yellow Room	Conan
S4	T5	2017-08-31	08:00:00	13:00:00	Yellow Room	Conan
S5	T1	2013-07-11	08:00:00	13:00:00	Red Room	Randall
S6	T4	2018-05-01	08:00:00	13:00:00	Blue Room	Ralph
S7	T3	2015-06-28	08:00:00	13:00:00	Yellow Room	Conan
S8	T1	2016-05-24	08:00:00	13:00:00	Red Room	Randall
S9	T3	2014-06-01	08:00:00	13:00:00	Yellow Room	Conan
NULL	NULL	NULL	NULL	NULL	NULL	NULL

After Update:

```
UPDATE training_session
SET
location = "Purple Room"
WHERE session_id = "S1"
```

session_id	training_id	date	start_time	end_time	location	instructor
S1	T1	2015-07-14	08:00:00	13:00:00	Purple Room	Randall
S10	T4	2017-06-22	08:00:00	13:00:00	Blue Room	Ralph
S2	T1	2014-05-29	08:00:00	13:00:00	Red Room	Randall
S3	T2	2013-12-01	08:00:00	13:00:00	Yellow Room	Conan
S4	T5	2017-08-31	08:00:00	13:00:00	Yellow Room	Conan
S5	T1	2013-07-11	08:00:00	13:00:00	Red Room	Randall
S6	T4	2018-05-01	08:00:00	13:00:00	Blue Room	Ralph
S7	T3	2015-06-28	08:00:00	13:00:00	Yellow Room	Conan
S8	T1	2016-05-24	08:00:00	13:00:00	Red Room	Randall
S9	T3	2014-06-01	08:00:00	13:00:00	Yellow Room	Conan
NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. **Delete.** Delete all attributes in training session where session ID is S5.

Before Deletion:

session_id	training_id	date	start_time	end_time	location	instructor
S1	T1	2015-07-14	08:00:00	13:00:00	Purple Room	Randall
S10	T4	2017-06-22	08:00:00	13:00:00	Blue Room	Ralph
S2	T1	2014-05-29	08:00:00	13:00:00	Red Room	Randall
S3	T2	2013-12-01	08:00:00	13:00:00	Yellow Room	Conan
S4	T5	2017-08-31	08:00:00	13:00:00	Yellow Room	Conan
S5	T1	2013-07-11	08:00:00	13:00:00	Red Room	Randall
S6	T4	2018-05-01	08:00:00	13:00:00	Blue Room	Ralph
S7	T3	2015-06-28	08:00:00	13:00:00	Yellow Room	Conan
S8	T1	2016-05-24	08:00:00	13:00:00	Red Room	Randall
S9	T3	2014-06-01	08:00:00	13:00:00	Yellow Room	Conan
NULL	NULL	NULL	NULL	NULL	NULL	NULL

After Deletion:

```
DELETE FROM training_session
WHERE session_id = "S5"
```

session_id	training_id	date	start_time	end_time	location	instructor
S1	T1	2015-07-14	08:00:00	13:00:00	Purple Room	Randall
S10	T4	2017-06-22	08:00:00	13:00:00	Blue Room	Ralph
S2	T1	2014-05-29	08:00:00	13:00:00	Red Room	Randall
S3	T2	2013-12-01	08:00:00	13:00:00	Yellow Room	Conan
S4	T5	2017-08-31	08:00:00	13:00:00	Yellow Room	Conan
S6	T4	2018-05-01	08:00:00	13:00:00	Blue Room	Ralph
S7	T3	2015-06-28	08:00:00	13:00:00	Yellow Room	Conan
S8	T1	2016-05-24	08:00:00	13:00:00	Red Room	Randall
S9	T3	2014-06-01	08:00:00	13:00:00	Yellow Room	Conan
NULL	NULL	NULL	NULL	NULL	NULL	NULL

SQL Queries for Business Metrics

1. Growth Rate

"HR Wizard" provides its client companies with the ability to calculate growth rate by returning the necessary data. See the graph below that shows the trend of Uno's growth for the years 2013 to 2018 using said data.

```
1  SELECT COUNT(employee_id) AS 'Incoming Headcount', YEAR(Start_date)
2  FROM employee
3  GROUP BY YEAR(start_date)
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

Incoming Headcount	YEAR(Start_date)
7	2013
5	2014
3	2015
2	2016
4	2017
7	2018

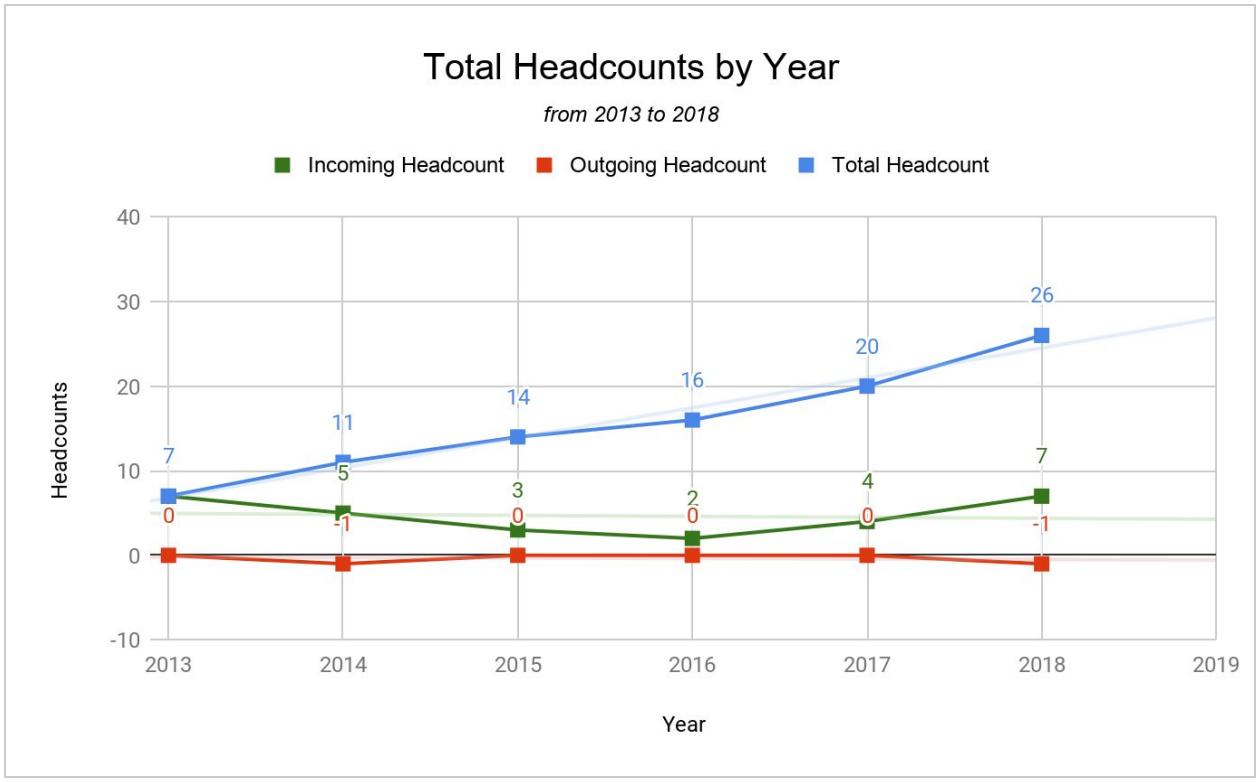
2. Attrition Rate

"HR Wizard" provides its client companies with the ability to calculate the attrition rate by returning the number of employees leaving the company per year. See the graph below that shows the trend of Uno's attrition rate the years 2013 to 2018 using said data.

```
1  SELECT COUNT(employee_id) AS 'Outgoing Headcount', YEAR(end_date)
2  FROM employee
3  GROUP BY YEAR(end_date)
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

Outgoing Headcount	YEAR(end_date)
26	NULL
1	2014
1	2018



3. Benefit Enrollment Rate

The percentage of Benefit Enrollment is calculated by:

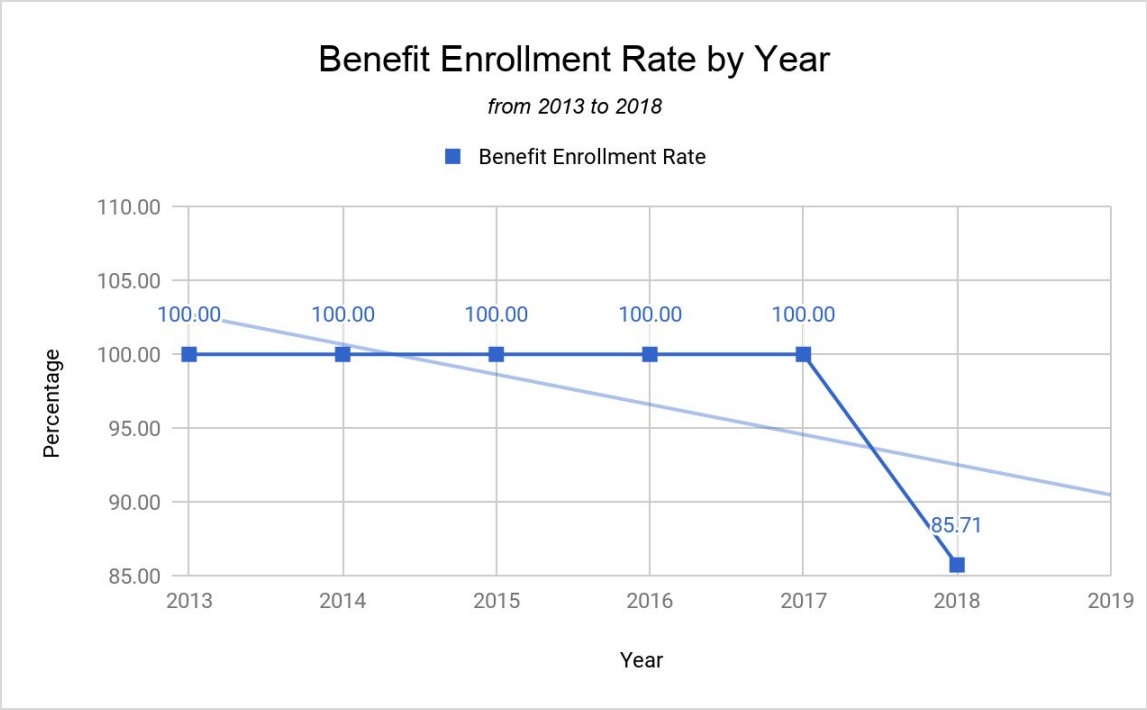
$$\frac{\#\ of\ employees\ enrolled\ in\ benefit(s)}{\#\ of\ employees} \times 100$$

This metric gives client companies the insight into what they can improve on, in terms of benefits offered, for attracting talents that they are looking for.

```
1  SELECT ((COUNT(b.benefit_id)/COUNT(e.employee_id))*100) AS
2  'Benefit Participation Rate', YEAR(e.start_date) AS 'Year'
3  FROM employee e LEFT OUTER JOIN benefits_enrolled b
4      ON e.employee_id = b.employee_id
5  GROUP BY YEAR(e.start_date)
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

Benefit Participation Rate	Year
100.0000	2013
100.0000	2014
100.0000	2015
100.0000	2016
100.0000	2017
85.7143	2018



4. Training Participation Rate

The percentage of Training Participation Rate is calculated by:

$$\frac{\#\ employees\ took\ at\ least\ one\ training\ course}{\#\ of\ employees} \times 100$$

This metric gives client companies the insight into what they can improve on in terms of training in order to achieve higher participation rate by their employees.

```
1  SELECT ((COUNT(t.enrollment_id)/COUNT(e.employee_id))*100) AS
2  'Training Participation Rate', YEAR(e.start_date) AS 'Year'
3  FROM employee e LEFT OUTER JOIN training_enrollment t
4      ON e.employee_id = t.employee_id
5      LEFT OUTER JOIN training_session ts
6      ON t.session_id = ts.session_id
7  GROUP BY YEAR(e.start_date)
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

Training Participation Rate	Year
57.1429	2013
80.0000	2014
100.0000	2015
100.0000	2016
100.0000	2017
14.2857	2018

Project Summary

Summarize your experience with this exercise

Overall, this was a challenging and time consuming project. The project required us to apply the concepts and materials that we learned in class. Through the process of completing this project, we get the opportunity to reinforce not only our understanding but also our appreciation for what we have learned.

What was the hardest part of this project?

The hardest part of the project was the creation of the UML model, which turned out to be quite complex. We had to edit and change it multiple times before arriving at the final model. The complexity of the UML has caused the creation of data difficult. We had to also come up with reasonable dependency for the various attributes i.e. start and end dates for an employee's stinct with the company, an employee's salary with his or her pay grade, and think about how all these different pieces fit together.

What problems did you run against in this project?

We encountered problems when quantifying our business metrics. Mistakenly, we did not think to make our metrics quantifiable in the database as we thought it would all come together in the end. To resolve the issue, we had to make many edits to the schema and data. This has also caused our project to be even more time consuming and more difficult to wrap our heads around.

` **How did you solve these problems?**

We were able to resolve most of our problems by getting valuable feedback from Professor Arwagal. On the other hand, with the addition of new attributes and data relations in our schema, we were able to resolve our problems eventually.

If you were to do this project again, what methodology would you follow?

To mediate the problems we had we would start the development of our UML earlier as it was a major pain point. Secondly, the ability to make our business metrics quantifiable posed to be a dramatic downfall to our overall planning. Therefore, we would consider this at an earlier stage of the project as well. With these adjustments we can make our project work as a smoother process with less unneeded complexity.

Suggestions for how to refine this project for the next class?

Due to the complexity of the project, suggested due dates spread across the quarter would be helpful. As this is also the first time that our team had undertaken a project of this caliber, sample projects for us to get an idea of the structure to follow would be greatly appreciated.