

ITCS 6160 (Fall 2017) Project

Internship System

Group 6:

Priyanka Sharma

Mingwei Qiang

Ren Zhang

Sri Sai Manish Kondapalli

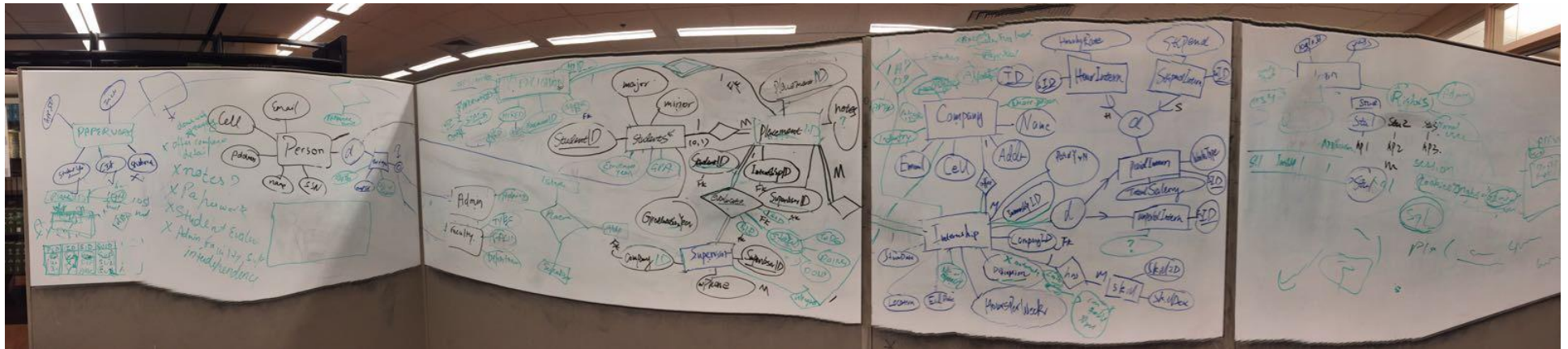
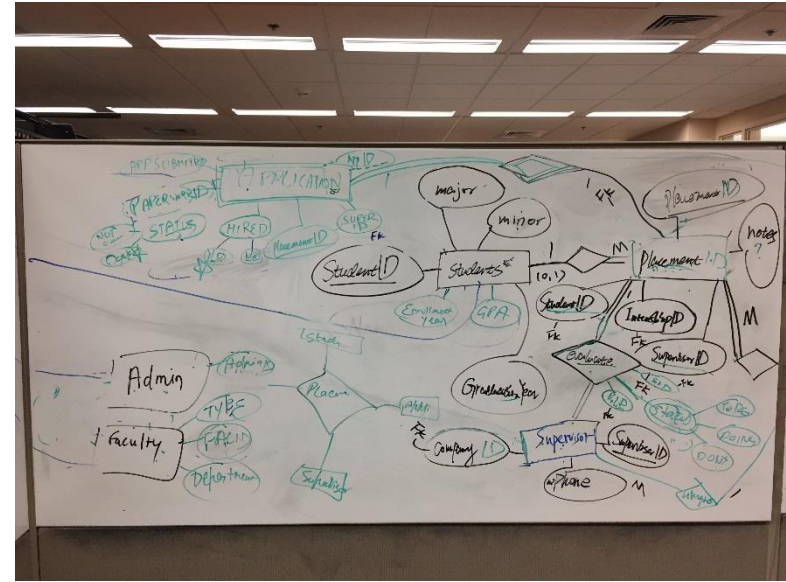
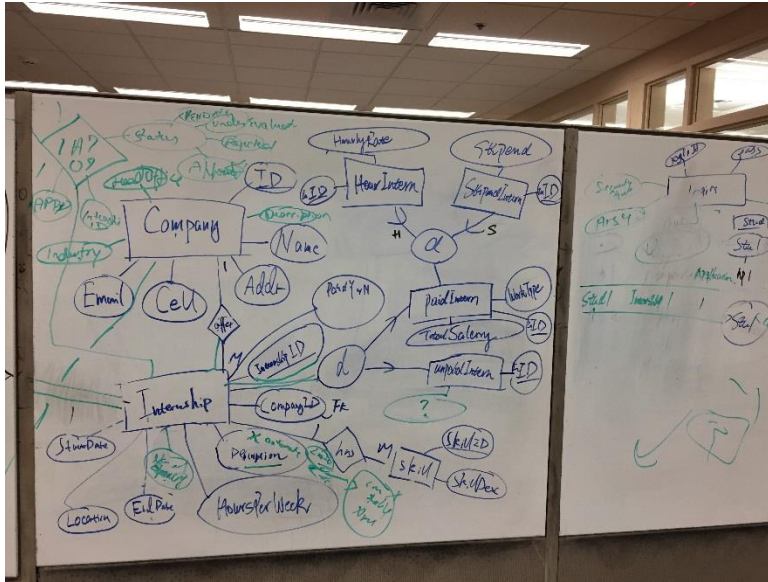
Sai-krishna Nadella

Dec 12th, 2017

Project Summary

- Discussion
- Create the EERD with Draw.IO
- Create Schema
- Create UML in MYSQL
- Add Constrains and references
- Forward engineer to SQL statements
- Debug, refine, and add functions on SQL statements
- Transform function and build interface on Web

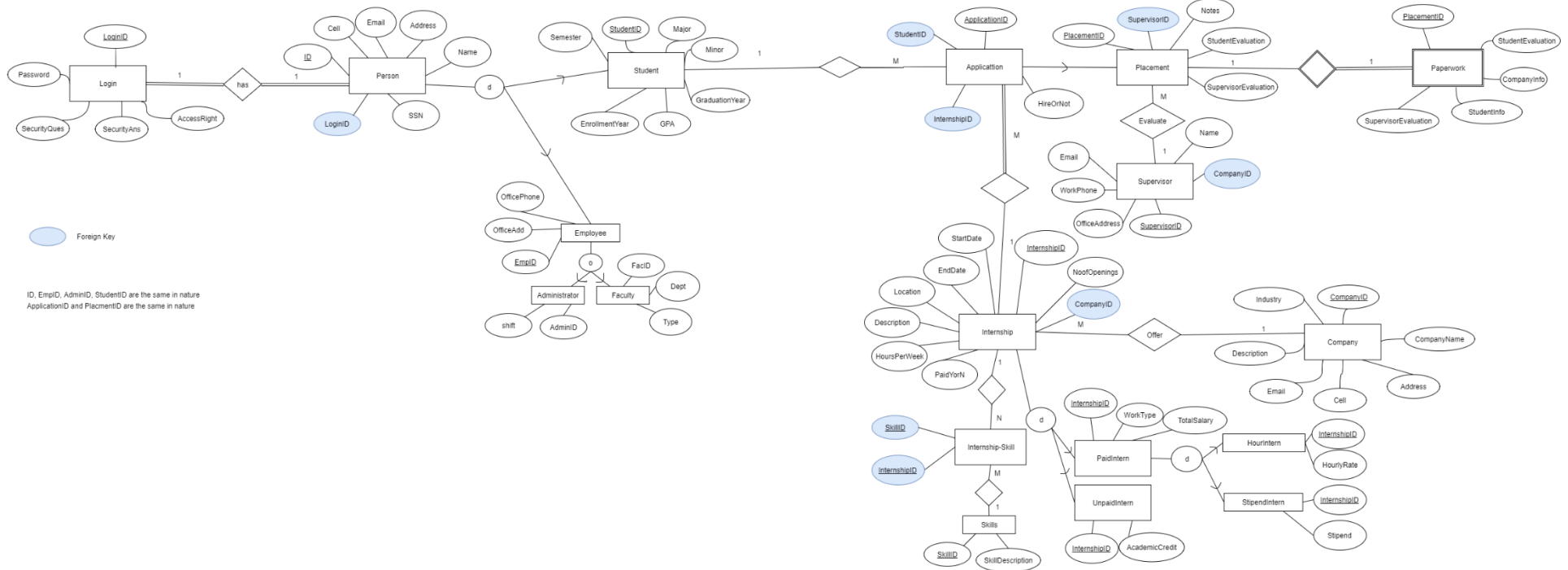
Discussion



Business Rules:

- Person is a general category for all people at the university, including students and employees which has 2 mutually exclusive specifications, faculties and administrators
- A person has one set of login information
- A company can offer one to many internships
- Each internship is associated with one and only one company
- An internship can receive infinite number of applications
- Each application is associated with one and only one internship
- A student can submit one to many applications
- Each application is associated with one and only one student
- Only applied internships can be placed
- A placement has a supervisor – the supervisor is from the same company the internship is offered by
- Each supervisor can evaluate one to many internships offered by his or her company
- A placement has one paperwork and each paperwork is associated with one and only one placement
- An internship needs one to many skills
- Each skill can be found in one to many internships
- An internship can be either paid or unpaid job
- Administrators have full access to the system
- Students, Faculty have read only access to companies, internships
- Only administrator can create, read, update/insert and delete internships, companies, placements

Description of EERD



Note:

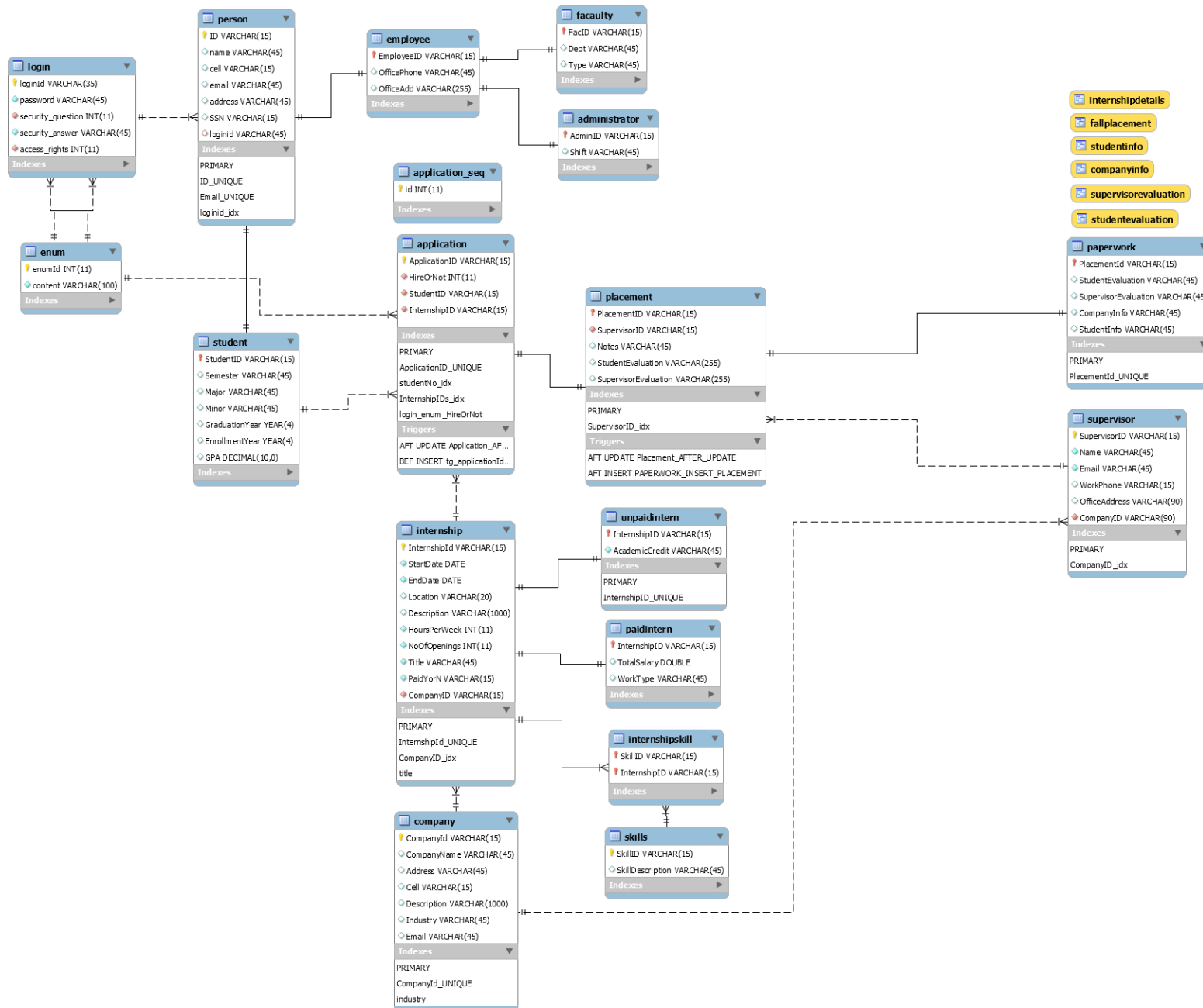
Each category under Person has a ID. The attributes ID in Person, StudentID, EmployeeID, FacID and AdminID are the same in nature.

ApplicationID and PlacementID are same in nature, PlacementID is a subset of ApplicationID

MySQL:

All tables are implemented in MySQL

Data Dictionary (meta data)



List and description of advanced SQL statements:

1. Stored procedure

Using stored procedure “computeCredits” to compute credits for each UnpaidInternship. The procedure will read EndDate, StartDate, HourPerWeek, Hours from Table Internship, and use formula “ $\text{credit} = \text{hours} * (\text{EndDate} - \text{StartDate}) / 7 * 0.13$ ” computing AcademicCredit, then update into Table unpaidIntern. Here is the code of this procedure.

```
-----  
  
USE `Internship_Inventory`;  
  
DROP procedure IF EXISTS `Internship_Inventory`.`computeCredits`;  
  
DELIMITER $$  
  
USE `Internship_Inventory`$$  
  
CREATE PROCEDURE `computeCredits`()  
  
BEGIN  
  
    DECLARE done INT DEFAULT FALSE;  
  
    DECLARE stdt, endt DATE;  
  
    DECLARE hours INT;  
  
    DECLARE id varchar(15);  
  
    DECLARE cur1 CURSOR FOR SELECT  
  
        INTERNSHIPID,STARTDATE,ENDDATE,HOURSPERWEEK  
  
        FROM INTERNSHIP;  
  
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;  
  
    OPEN cur1;  
  
read_loop: LOOP  
  
        FETCH cur1 INTO id, stdt, endt, hours;
```

```

        IF done THEN
        LEAVE read_loop;

        END IF;

        UPDATE unpaidIntern SET AcademicCredit = hours*(DATEDIFF(endt,stdt)/7)*0.13

        WHERE internshipid=id;

    END LOOP;

    CLOSE cur1;

END$$

DELIMITER ;

```

This is how this procedure is implemented in database. First we insert two unpaidIntern data into table with null of AcademicCredit's value. Then call the Stored procedure "computeCredits" to update the field's value.

```

START TRANSACTION;

USE `internship_inventory`;

INSERT INTO `internship_inventory`.`unpaidintern` (`InternshipID`, `AcademicCredit`) VALUES ('I301', null);

INSERT INTO `internship_inventory`.`unpaidintern` (`InternshipID`, `AcademicCredit`) VALUES ('I401', null);

call computeCredits();

COMMIT;

```

Here is the table displays before and after call "computeCredits".

	InternshipID	AcademicCredit
	I301	NULL
	I401	NULL
	NULL	NULL

	InternshipID	AcademicCredit
	I301	67.78571428460
	I401	51.25714285380
	NULL	NULL

2. Trigger

We have two set of triggers.

First set of triggers is for inserting and updating. In our schema, paperwork is the weak entity of placement, and placement is the subset of application. It makes placementId and applicationId need to be updated in three tables.

Trigger “Application_AFTER_UPDATE” is activated after updating Table application, if hireOrNot status is 11 (which means hired), then a new data is inserted into Table paperwork.

```
-----  
USE `Internship_Inventory`$$  
DROP TRIGGER IF EXISTS `Internship_Inventory`.`Application_AFTER_UPDATE` $$  
USE `Internship_Inventory`$$  
CREATE DEFINER = CURRENT_USER TRIGGER `Internship_Inventory`.`Application_AFTER_UPDATE` AFTER UPDATE ON `Application` FOR EACH ROW  
BEGIN  
    if `old`.`hireornot` = 11 THEN  
        insert into paperwork (placementid) values (`placementid`);  
    end if;  
END$$
```

Tigger “Placement_AFTER_UPDATE” and “PAPERWORK_INSERT_PLACEMENT” are activated after updating and inserting data into Table placement, and the placementID is updated or inserted into Table paperwork.

```
-----  
USE `Internship_Inventory`$$  
DROP TRIGGER IF EXISTS `Internship_Inventory`.`Placement_AFTER_UPDATE` $$  
USE `Internship_Inventory`$$  
CREATE DEFINER = CURRENT_USER TRIGGER `Internship_Inventory`.`Placement_AFTER_UPDATE` AFTER UPDATE ON `Placement` FOR EACH ROW  
BEGIN
```

```
        insert into paperwork (placementid) values (old.placementid);  
END$$
```

```
USE `Internship_Inventory`$$
```

```
DROP TRIGGER IF EXISTS `Internship_Inventory`.`PAPERWORK_INSERT_PLACEMENT` $$
```

```
USE `Internship_Inventory`$$
```

```
CREATE DEFINER = CURRENT_USER TRIGGER `Internship_Inventory`.`PAPERWORK_INSERT_PLACEMENT`
```

```
AFTER INSERT ON `placement` FOR EACH ROW
```

```
BEGIN
```

```
        insert into paperwork (placementid) values (new.placementid);
```

```
END$$
```

Second set of triggers is for automatically adding applicationID with a certain format when a new data is inserted into table application. We need the function that after some student fill in information and send application, a new data is inserted without applicaitonID input. First, we create a table “application_seq” to generate autoincrement number. Then we concat the information of insertion and new autoincrement number to an applicationID. This applicationID will be added before insertion.

```
DELIMITER ;
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`application_seq` ;
```

```
CREATE TABLE application_seq
```

```
(
```

```
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY
```

```
);
```

```
DELIMITER $$
```

```
USE `Internship_Inventory`$$
```

```
DROP TRIGGER IF EXISTS `Internship_Inventory`.`tg_applicationId_insert` $$
```

```
CREATE TRIGGER tg_applicationId_insert
```

```
BEFORE INSERT ON application
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO application_seq VALUES (NULL);
```

```
    SET NEW.applicationId = CONCAT('A',SUBSTRING(NEW.InterNshipID, 2, length(NEW.InterNshipID)-1),LAST_INSERT_ID());
```

```
END$$
```

3. Delete and Update

Delete data where applicationID='A2012' to see how it works in table application. Further see how it works in table placement since table placement has placementID as foreign key of applicationID.

```
-----  
SELECT * from application;  
  
DELETE FROM application  
  
where applicationID='A2012';  
  
SELECT * from application;  
  
SELECT * from placement;  
  
-----  
-----
```

Table application before deletion.

	ApplicationID	HireOrNot	StudentID	InternshipID
	A1011	10	10004	I101
	A2012	11	10005	I201
	A3013	11	10007	I301
	A4014	12	10006	I401
	A4025	12	10008	I402

Table application after deletion. Data with applicationID 'A2012' is deleted.

	ApplicationID	HireOrNot	StudentID	InternshipID
	A1011	10	10004	I101
	A3013	11	10007	I301
	A4014	12	10006	I401
	A4025	12	10008	I402

Table placement after deletion. Data with placementID 'A2012' is deleted as well.

	PlacementID	SupervisorID	Notes
	A1011	S10001	Pending
	A2011	S10002	Pending
	A3013	S10002	Pending

Update data where applicationID='A4014' from value 12 to 10.

```
SELECT * from application;
```

```
UPDATE application
```

```
SET HireOrNot = 10
```

```
WHERE applicationID = 'A4014';
```

```
SELECT * from application;
```

-----Before updating. The value of field HireOrNot in third row has been updated to 10.

	ApplicationID	HireOrNot	StudentID	InternshipID
	A1011	10	10004	I101
	A3013	11	10007	I301
	A4014	12	10006	I401
	A4025	12	10008	I402

After updating.

	ApplicationID	HireOrNot	StudentID	InternshipID
	A1011	10	10004	I101
	A3013	11	10007	I301
	A4014	10	10006	I401
	A4025	12	10008	I402

4. Views

We created 6 views. The first two is the group-specific to.... , the other four corresponds to the reports of paperwork.

```
-- View-1 `Internship_Inventory`.`internshipDetails`
```

```
DROP VIEW IF EXISTS `Internship_Inventory`.`internshipDetails` ;
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`internshipDetails`;
```

```
USE `Internship_Inventory`;
```

```
CREATE OR REPLACE VIEW `internshipDetails` AS
```

```
select distinct INTE.internshipID, INTE.TITLE, INTE.DESCRPTION, skills.skillDescription, INTE.PAIDYORN, INTE.NOOFOPENINGS,CO.COMPANYNAME, CO.INDUSTRY
FROM COMPANY AS CO, INTERNSHIP AS INTE, skills, internshipskill
```

```
where INTE.COMPANYID=CO.COMPANYID and INTE.internshipID = internshipskill.InternshipID and internshipskill.SkillID=skills.SkillID;
```

Result:

internshipID	TITLE	DESCRIPTION	skillDescription	PAIDYORN	NOOFOPENINGS	COMPANYNAME	INDUSTRY
I101	Data Analytics	The successful candidate will turn dat...	Java	Y	2	Walmart	Retail
I201	Marketing	Marketing Assistant responsibilities incl...	Java	Y	3	Exxon Mobil Corporation	Petroleum refining
I201	Marketing	Marketing Assistant responsibilities incl...	SOL	Y	3	Exxon Mobil Corporation	Petroleum refining
I301	Developer	Java developer roles and responsibiliti...	Python	N	1	TOYOTA MOTOR	automotive manufacturers
I401	Sales	Sales assistant duties include selling. r...	Java	N	5	Apple Inc.	technology
I401	Sales	Sales assistant duties include selling. r...	Excel	N	5	Apple Inc.	technology

Description:

This is the view to display details of internships. It queries data from four tables, company, internship, skills, and internshipskills and display the result of joining sets.

```
-- View-2 `Internship_Inventory`.`FallPlacement`
```

```
-- All placements for a particular time frame-Fall
```


Codes:

```
DROP VIEW IF EXISTS `Internship_Inventory`.`FallPlacement` ;  
DROP TABLE IF EXISTS `Internship_Inventory`.`FallPlacement`;  
USE `Internship_Inventory`;  
CREATE OR REPLACE VIEW `FallPlacement` AS  
SELECT distinct placement.placementID, person.name as studentName, company.CompanyName,  
supervisor.name as supervisorName, application.internshipId, notes  
FROM Placement, application, Person, supervisor, internship, company  
WHERE PlacementID IN (SELECT ApplicationID FROM Application  
WHERE application.StudentID IN (SELECT StudentID FROM student WHERE semester='Fall'))  
and placementID = application.applicationID and application.studentID = person.ID  
and application.internshipID = internship.internshipID and internship.companyID = company.companyID  
and placement.supervisorID = supervisor.SupervisorID;
```

Result:

	placementID	studentName	CompanyName	supervisorName	internshipId	notes
	A2012	DanielBrown	Exxon Mobil Corporation	Mvriam Oneill	I201	Pending
	A3013	StephenYoung	TOYOTA MOTOR	Johanne Rawlins	I301	Pending

Description:

This view includes all Placement information for Fall semester. It is designed for faculties in University Career Center, who care about the placements in a particular semester. The number of placements in one semester can serve as one of the criteria to measure how much effort that the Career Center has made to help students. In addition, the relation between semester(Fall, Spring or Summer), Company and Internship are important information to generate suggestions on job application timing to students in the future. Last, faculties can upload notes to each placement by the end of semester.

```
-- View-3 `Internship_Inventory`.`StudentInfo`
```

```
-- All student information
```

```
Codes:
```

```
DROP VIEW IF EXISTS `Internship_Inventory`.`StudentInfo` ;
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`StudentInfo`;
```

```
USE `Internship_Inventory`;
```

```
CREATE OR REPLACE VIEW `StudentInfo` AS
```

```
SELECT s.studentID, p.Name, p.cell, p.email, p.address, p.ssn, s.Semester, s.Major, s.Minor, s.GraduationYear, s.EnrollmentYear, s.GPA
```

```
FROM person p
```

```
RIGHT JOIN students on p.ID=s.studentID;
```

```
WHERE s.studentID IN (SELECT studentID FROM application WHERE applicationID IN (SELECT PlacementID FROM Placement));
```

```
Result:
```

studentID	Name	cell	email	address	ssn	Semester	Major	Minor	GraduationYear	EnrollmentYear	GPA
10004	SheliaCurtis	(671) 925-1352	crandall@sbcglobal.net	241 Indian Spring St. Pittsburg, CA 94565	527-80-4977	Spring	Math	NULL	2020	2016	4
10005	DanielBrown	(732) 925-1352	dbrown@sbcglobal.net	3336 Indian Spring New Brunswick 08933	511-80-4934	Fall	Art	NULL	2018	2016	3
10007	StephenYoung	(212) 925-1352	syoung@sbcglobal.net	70 Washington Square South New York, NY 10003	533-80-4912	Fall	Finance	NULL	2022	2018	3

```
Description:
```

This view shows all information of student who were hired. This is combined information grabbed from Student table and Person table. The information is for the paperwork “Student Information”.

```

-----
-- View-4 `Internship_Inventory`.`CompanyInfo`
-- All company information including internship they provide
-----

```

```

DROP VIEW IF EXISTS `Internship_Inventory`.`CompanyInfo` ;
DROP TABLE IF EXISTS `Internship_Inventory`.`CompanyInfo`;
USE `Internship_Inventory`;
CREATE OR REPLACE VIEW `CompanyInfo` AS
SELECT c.companyid, c.companyname, c.address, c.cell, c.description,c.email, industry, i.title, internshipid, noofopenings
FROM company c
LEFT JOIN internship i ON c.companyid=i.companyid;

```

Result:

companyid	companyname	address	cell	description	email	industry	title	internshipid	noofopenings
C1	Walmart	702 Sw 8th St, Bentonville, AR	(479) 273-4000	Wal-Mart Stores is an irr	www.walmart.com	Retail	Data Analytics	I101	2
C2	Exxon Mobil Corporation	4045 Scenic Hwy, Baton Rouge, LA	(985) 359-8519	Some telephone number	www.exxonmobil.com	Petroleum refining	Marketing	I201	3
C3	TOYOTA MOTOR	1, Toyotacho, Toyota, Aichi,Japan	81-565282121	Toyota Motor, among th	www.toyota.com	automotive manufacturers	Developer	I301	1
C4	Apple Inc.	1 Infinite Loop, Cupertino, CA	(408) 996-1010	Ask Siri to name the mos	www.apple.com	technology	Sales	I401	5

Description:

This view includes all information of companies which provide internship opportunities and the type of work (indicated by title) they provide. This provides students with some ideas of what companies they can follow up based on their career goals and what type of jobs they can consider if they can dream companies.

```

-----
-- View-5 `Internship_Inventory`.`SupervisorEvaluation`
-- Supervisor's evaluation for company to review
-----

DROP VIEW IF EXISTS `Internship_Inventory`.`SupervisorEvaluation` ;
DROP TABLE IF EXISTS `Internship_Inventory`.`SupervisorEvaluation`;
USE `Internship_Inventory`;
CREATE OR REPLACE VIEW `SupervisorEvaluation` AS
SELECT placement.placementid, person.name as 'student_name', internship.title, semester, companyname, supervisor.supervisorid, supervisor.name as
'supervisor_name', Placement.SupervisorEvaluation
FROM placement, person, internship, company, supervisor, application, student
WHERE placement.SupervisorEvaluation<> "AND placement.placementid=application.applicationID AND application.studentID=student.studentID AND
student.studentID=person.ID AND
application.internshipID=internship.internshipID AND internship.companyID=company.companyID AND
placement.supervisorID=supervisor.supervisorID
GROUP BY companyname;

```

Results:

placementid	student_name	title	semester	companyname	supervisorid	supervisor_name	SupervisorEvaluation
A1011	Shelia@urtis	Data Analytics	Spring	Walmart	S10001	Myriam Oneill	Creative

Description:

This view includes the non-null SupervisorEvaluation and the basic placement, internship and company associated to the evaluation. This view is for the report “SupervisorEvaualtion”.

```

-----
-- View-6 `Internship_Inventory`.`StudentEvaluation`
-- Student's evaluation on their completed internship
-----

DROP VIEW IF EXISTS `Internship_Inventory`.`StudentEvaluation` ;
DROP TABLE IF EXISTS `Internship_Inventory`.`StudentEvaluation`;
USE `Internship_Inventory`;
CREATE OR REPLACE VIEW `StudentEvaluation` AS
SELECT companyname, title, semester, person.name, studentevaluation
FROM company, internship, student, person, placement,application
WHERE placement.studentevaluation<>" AND placement.placementID=application.applicationID AND
application.studentID=student.studentID AND student.studentID=person.ID AND
application.internshipID=internship.internshipID AND internship.companyID=company.companyID
GROUP BY companyname;
SELECT * FROM StudentEvaluation;

```

Results:

companyname	title	semester	name	studentevaluation
Walmart	Data Analytics	Spring	Shelia@urtis	Pleasant workplace

Description:

This view includes all non-null StudentEvaluation and information of company, title, semester and studentname associated to this evaluation. This view is for the report “StudentEvaluation”.

5. Transaction

This transaction is for displaying insert a new data and rollback.

Before transaction.

ApplicationID	HireOrNot	StudentID	InternshipID
A1011	10	10004	I101
A2012	11	10005	I201
A3013	11	10007	I301
A4014	12	10006	I401
A4025	12	10008	I402
NULL	NULL	NULL	NULL

START TRANSACTION;

USE `internship_inventory`;

INSERT INTO `internship_inventory`.`application` (`HireOrNot`, `StudentID`, `InternshipID`) VALUES (12, '10006', 'I101');

rollback;

Because the real transaction is not commit and rollback instead, the results is:

ApplicationID	HireOrNot	StudentID	InternshipID
A1011	10	10004	I101
A2012	11	10005	I201
A3013	11	10007	I301
A4014	12	10006	I401
A4025	12	10008	I402
NULL	NULL	NULL	NULL

This transaction doesn't write into database..

6. Index

In table person, the primary ID is indexed for sure. And because the table stored frequently information for login, the other two attributes email (as login name) and loginid(as alternative login name) are indexed as well. It will increase query efficiency while user input their email or loginId. The database system don't have to scan full data, but read the address from index instead.

DROP TABLE IF EXISTS `internship_inventory`.`person` ;

CREATE TABLE IF NOT EXISTS `internship_inventory`.`person` (

 `ID` VARCHAR(15) NOT NULL,

 `name` VARCHAR(45) NULL DEFAULT NULL,

 `cell` VARCHAR(15) NULL DEFAULT NULL,

 `email` VARCHAR(45) NULL DEFAULT NULL,

 `address` VARCHAR(45) NULL DEFAULT NULL,

 `SSN` VARCHAR(15) NULL DEFAULT NULL,

 `loginid` VARCHAR(45) NULL DEFAULT NULL,

 PRIMARY KEY (`ID`),

 UNIQUE INDEX `ID_UNIQUE` (`ID` ASC),

 UNIQUE INDEX `Email_UNIQUE` (`Email` ASC),

 INDEX `loginid_idx` (`LoginID` ASC),

 CONSTRAINT `loginid`

 FOREIGN KEY (`loginid`)

 REFERENCES `internship_inventory`.`login` (`loginid`)

 ON DELETE NO ACTION

 ON UPDATE NO ACTION)-----

7. UI Screenshots



localhost:8080/internship_system

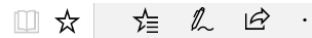


Internship Management System

[Click To Login](#)



localhost:8080/internship_system/submitHome.html



User Name: 10001

Password:

Submit

[Continue to home](#)

FALL PLACEMENT

PARTNER COMPANIES

ACCOUNT DETAILS

logout

Internship
Opportunities
Search Job Title

[Edit
Internships
Edit
Applications](#)

Title	Description	Paid	No Of Openings	Company Name	Industry	More...
Data Analytics	The successful candidate will turn data into information, information into insight and insight into business decisions. Data analyst responsibilities include conducting full lifecycle analysis to include requirements, activities and design. Data analysts will develop analysis and reporting capabilities. They will also monitor performance and quality control plans to identify improvements.		2	Walmart	Retail	Details
Marketing	Marketing Assistant responsibilities include assisting in organizing campaigns and developing marketing strategies. This work will be a critical factor for the smooth operation of the Marketing department and the attainment of its goals, as well as for the long-term growth of the company.		3	Exxon Mobil Corporation	Petroleum refining	Details
Marketing	Marketing Assistant responsibilities include assisting in organizing campaigns and developing marketing strategies. This work will be a critical factor for the smooth operation of the Marketing department and the attainment of its goals, as well as for the long-term growth of the company.		3	Exxon Mobil Corporation	Petroleum refining	Details
Developer	Java developer roles and responsibilities include managing Java/Java EE application development while providing expertise in the full software development lifecycle, from concept and design to testing.		1	TOYOTA MOTOR	automotive manufacturers	Details
Sales	Sales assistant duties include selling, restocking and merchandising. The goal is to provide high class customer service and to increase company?s growth and revenue through sales maximisation.		5	Apple Inc.	technology	Details
Sales	Sales assistant duties include selling, restocking and merchandising. The goal is to provide high class customer service and to increase company?s growth and revenue through sales maximisation.		5	Apple Inc.	technology	Details

[Continue to home](#)

FALL PLACEMENT

PARTNER COMPANIES

ACCOUNT DETAILS

logout

[Edit Internships](#)
[Edit Applications](#)

Internship Opportunities

Search Job Title

Title	Description	Paid	No Of Openings	Company Name	Industry	More...
Marketing	Marketing Assistant responsibilities include assisting in organizing campaigns and developing marketing strategies. This work will be a critical factor for the smooth operation of the Marketing department and the attainment of its goals, as well as for the long-term growth of the company.		3	Exxon Mobil Corporation	Petroleum refining	Details
Marketing	Marketing Assistant responsibilities include assisting in organizing campaigns and developing marketing strategies. This work will be a critical factor for the smooth operation of the Marketing department and the attainment of its goals, as well as for the long-term growth of the company.		3	Exxon Mobil Corporation	Petroleum refining	Details

Created For:ITCS 6160

Contact information: psharm16@uncc.edu.



Internship Management System

[Continue to home](#)

Complete Details

Start Date :2017-05-01 End Date :2017-10-01 Location : New York Description :Marketing Assistant responsibilities include assisting in organizing campaigns and developing marketing strategies. This work will be a critical factor for the smooth operation of the Marketing department and the attainment of its goals, as well as for the long-term growth of the company. Hours Per Week :40 No Of Openings :3 Title :Marketing Company Name :Exxon Mobil Corporation Address :4045 Scenic Hwy, Baton Rouge, LA Cell :(985) 359-8519 Description :Marketing Assistant responsibilities include assisting in organizing campaigns and developing marketing strategies. This work will be a critical factor for the smooth operation of the Marketing department and the attainment of its goals, as well as for the long-term growth of the company. Industry :Petroleum refining Email :www.exxonmobil.com Work Type :Onsite Total Salary :30000.0 Skills :Java SQL

Created For:ITCS 6160

Contact information: psharm16@uncc.edu.

s

[Continue to home](#)

Partner Companies

Company Name	Address	Cell	Industry	Email	Title	Description	No Of Openings
Walmart	702 Sw 8th St, Bentonville, AR	(479) 273-4000	Retail	www.walmart.com	Data Analytics	Wal-Mart Stores is an irresistible (or at least unavoidable) retail force that has yet to meet any immovable objects. The world's largest company by revenue and bigger than Europe's Carrefour, Metro AG, and Tesco combined, Walmart is the world's #1 retailer with more than 2.3 million employees. In the US, Wal-Mart operates more than 5,300 stores, including about 4,570 Wal-Mart stores and 655 Sam's Club warehouses, and a growing number of smaller format stores. The company's faster growing international division (26% of sales) numbers more than 6,100 locations; Wal-Mart is the #1 retailer in Canada and Mexico and has operations in Asia (where it owns a 95% stake in Japanese retailer SEIYU), Africa, Europe, and Latin America.	2
Exxon Mobil Corporation	4045 Scenic Hwy, Baton Rouge, LA	(985) 359-8519	Petroleum refining	www.exxonmobil.com	Marketing	Some telephone numbers on the Hoover's site may be on a country's do not call or do not contact list including, but not limited to, the United Kingdom's CTPS or TPS registers. It is a legal requirement that companies do not make sales or marketing calls to registered numbers. These are central opt out registers whereby corporate subscribers and individuals can register their preference not to receive unsolicited sales and marketing telephone calls. By using the information provided on the Hoover's sites, as the direct marketer you represent and warrant that you will use such information in compliance with all applicable local, state, national or international laws and regulations, including any local do not call registers or marketing regulations, and agree to defend, indemnify and hold harmless Dun & Bradstreet and each of its affiliates in the event your use violates such laws and regulations.	3
TOYOTA MOTOR	1, Toyotacho, Toyota, Aichi, Japan	+81-565282121	automotive manufacturers	www.toyota.com	Developer	Toyota Motor, among the world's largest automotive manufacturers by revenue, designs and manufactures a diverse product line-up that ranges from subcompacts to luxury and sports vehicles to SUVs, trucks, minivans, and buses. Its vehicles are produced either with combustion or hybrid engines, as with the Prius. Toyota's subsidiaries also manufacture vehicles: Daihatsu Motor produces mini-vehicles, while Hino Motors produces trucks and buses. Additionally, Toyota makes automotive parts for its own use and for sale to others. Popular models include the Camry, Corolla, Land Cruiser, and luxury Lexus line, as well as the Tundra truck.	1
Apple Inc.	1 Infinite Loop, Cupertino, CA	(408) 996-1010	technology	www.apple.com	Sales	Ask Siri to name the most successful company in the world and it might respond: Apple. In terms of profit, revenue, market capitalization, and consumer cachet, it certainly ranks right up there. The iPhone, in its ninth year and seventh generation, has been the company's golden goose, although the aging product may be losing a little of its luster. In addition to the iPhone, other familiar Apple products and services include MacBook computers and iPad tablets, as well as iTunes, the App store, and Apple Music. Primarily a consumer-oriented company, Apple has inked alliances with IBM and Cisco Systems to deepen its penetration of the enterprise market. About 60% of revenue comes from outside the Americas.	5



Internship Management System

[Click To Login](#)



Internship Management System

[Continue to home](#)

Complete Details

Click Apply to be considered for Job Position

Start Date :2018-07-01 End Date :2019-07-01 Location :Chicago Description :Java developer roles and responsibilities include managing Java/Java EE application development while providing expertise in the full software development lifecycle, from concept and design to testing. Hours Per Week :10 No Of Openings :1 Title :Developer Company Name :TOYOTA MOTOR Address :1, Toyotacho, Toyota, Aichi,Japan Cell :[+81-565282121](tel:+81-565282121)
Description :Java developer roles and responsibilities include managing Java/Java EE application development while providing expertise in the full software development lifecycle, from concept and design to testing.
Industry :automotive manufacturers Email :www.toyota.com Academic Credit :67.78571428460 Skills :Python

[Apply](#)

Created For:ITCS 6160

Contact information: psharm16@uncc.edu.



Internship Management System

Applied

[Continue to home](#)

Created For:ITCS 6160

Contact information: psharm16@uncc.edu.



Internship Management System

[Continue to home](#)

Placement View

Student Name	Company Name	Supervisor Name	Notes
Daniel Brown	Exxon Mobil Corporation	Myriam Oneill	Pending
Stephen Young	TOYOTA MOTOR	Johanne Rawlings	Pending

Created For:ITCS 6160

Contact information: psharm16@uncc.edu.

Appendix:

dump of SQL code and PHP/other code

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-----

-- Schema Internship_Inventory

-----

DROP SCHEMA IF EXISTS `Internship_Inventory` ;

-----

-- Schema Internship_Inventory

-----

CREATE SCHEMA IF NOT EXISTS `Internship_Inventory` DEFAULT CHARACTER SET utf8;

USE `Internship_Inventory` ;

-----

-- Table `Internship_Inventory`.`Login`

-----

DROP TABLE IF EXISTS `internship_inventory`.`login` ;

CREATE TABLE IF NOT EXISTS `internship_inventory`.`login` (

  `loginId` VARCHAR(35) NOT NULL,

  `password` VARCHAR(45) NOT NULL,
```

```

`security_question` int NOT NULL,

`security_answer` VARCHAR(45) NOT NULL,

`access_rights` int NOT NULL,

PRIMARY KEY (`LoginId`),

UNIQUE INDEX `LoginId_UNIQUE` (`LoginId` ASC),

CONSTRAINT `login_enum_security_question`

    FOREIGN KEY (`security_question`)

    REFERENCES `internship_inventory`.`enum` (`enumID`),

CONSTRAINT `login_enum_access_rights`

    FOREIGN KEY (`access_rights`)

    REFERENCES `internship_inventory`.`enum` (`enumID`)

)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

```

```

-- Table `Internship_Inventory`.`Person`

```

```

DROP TABLE IF EXISTS `internship_inventory`.`person` ;

CREATE TABLE IF NOT EXISTS `internship_inventory`.`person` (

    `ID` VARCHAR(15) NOT NULL,

    `name` VARCHAR(45) NULL DEFAULT NULL,

    `cell` VARCHAR(15) NULL DEFAULT NULL,

    `email` VARCHAR(45) NULL DEFAULT NULL,

```

```
`address` VARCHAR(45) NULL DEFAULT NULL,  
`SSN` VARCHAR(15) NULL DEFAULT NULL,  
`loginid` VARCHAR(45) NULL DEFAULT NULL,  
PRIMARY KEY (`ID`),  
UNIQUE INDEX `ID_UNIQUE` (`ID` ASC),  
UNIQUE INDEX `Email_UNIQUE` (`Email` ASC),  
INDEX `loginid_idx` (`LoginID` ASC),  
CONSTRAINT `loginid`  
  FOREIGN KEY (`loginid`)  
  REFERENCES `internship_inventory`.`login` (`loginid`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;
```

```
-- -----  
-- Table `Internship_Inventory`.`Employee`  
-- -----
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`Employee` ;  
CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`Employee` (  
  `EmployeeID` VARCHAR(15) NOT NULL,  
  `OfficePhone` VARCHAR(45) NULL,  
  `OfficeAdd` VARCHAR(255) NULL,  
  PRIMARY KEY (`EmployeeID`),  
  CONSTRAINT `EmployeeID`
```

```
FOREIGN KEY (`EmployeeID`)
REFERENCES `Internship_Inventory`.`Person` (`ID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;
```

```
-- Table `Internship_Inventory`.`Administrator`
```

```
-----

DROP TABLE IF EXISTS `Internship_Inventory`.`Administrator` ;

CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`Administrator` (

`AdminID` VARCHAR(15) NOT NULL,

`Shift` VARCHAR(45) NULL,

PRIMARY KEY (`AdminID`),

CONSTRAINT `AdminID`

FOREIGN KEY (`AdminID`)

REFERENCES `Internship_Inventory`.`Employee` (`EmployeeID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;
```

```
-- Table `Internship_Inventory`.`Faculty`
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`Faculty` ;

CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`Faculty` (

  `FacID` VARCHAR(15) NOT NULL,

  `Dept` VARCHAR(45) NULL,

  `Type` VARCHAR(45) NULL,

  PRIMARY KEY (`FacID`),

  CONSTRAINT `FacID`

    FOREIGN KEY (`FacID`)

      REFERENCES `Internship_Inventory`.`Employee` (`EmployeeID`)

    ON DELETE CASCADE

    ON UPDATE CASCADE)

ENGINE = InnoDB;
```

```
-- Table `Internship_Inventory`.`Student`
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`Student` ;

CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`Student` (

  `StudentID` VARCHAR(15) NOT NULL,

  `Semester` VARCHAR(45) NULL,

  `Major` VARCHAR(45) NULL,

  `Minor` VARCHAR(45) NULL,

  `GraduationYear` YEAR NULL,

  `EnrollmentYear` YEAR NULL,

  `GPA` DECIMAL NULL,
```

```
PRIMARY KEY (`StudentID`),  
CONSTRAINT `student_id`  
FOREIGN KEY (`StudentID`)  
REFERENCES `Internship_Inventory`.`Person` (`ID`)  
ON DELETE NO ACTION  
ON UPDATE CASCADE)  
ENGINE = InnoDB;
```

```
-- Table `Internship_Inventory`.`Company`  
-----
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`Company` ;  
CREATE TABLE IF NOT EXISTS `internship_inventory`.`company` (  
  `CompanyId` VARCHAR(15) NOT NULL,  
  `CompanyName` VARCHAR(45) NULL DEFAULT 'Anonymous',  
  `Address` VARCHAR(45) NULL,  
  `Cell` VARCHAR(15) NULL DEFAULT NULL,  
  `Description` VARCHAR(1000) NULL,  
  `Industry` VARCHAR(45) NULL,  
  `Email` VARCHAR(45) NULL,  
  PRIMARY KEY (`CompanyId`),  
  INDEX `industry` USING BTREE (`Industry` ASC),  
  UNIQUE INDEX `CompanyId_UNIQUE` (`CompanyId` ASC))  
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = utf8;
```

```
-- Table `Internship_Inventory`.`Skills`
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`Skills` ;
```

```
CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`Skills` (
```

```
  `SkillID` VARCHAR(15) NOT NULL,
```

```
  `SkillDescription` VARCHAR(45) NULL,
```

```
  PRIMARY KEY (`SkillID`))
```

```
ENGINE = InnoDB;
```

```
-- Table `Internship_Inventory`.`InternshipSkill`
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`InternshipSkill` ;
```

```
CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`InternshipSkill` (
```

```
  `SkillID` VARCHAR(15) NOT NULL,
```

```
  `InternshipID` VARCHAR(15) NOT NULL,
```

```
  PRIMARY KEY (`SkillID`, `InternshipID`),
```

```
  INDEX `InternshipID_idx` (`InternshipID` ASC),
```

```
  CONSTRAINT `SkillId`
```

```
    FOREIGN KEY (`SkillID`)
```

```
      REFERENCES `Internship_Inventory`.`Skills` (`SkillID`)
```



```
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `InternshipSkill_Internship_InternshipID`
FOREIGN KEY (`InternshipID`)
REFERENCES `Internship_Inventory`.`Internship` (`InternshipId`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table `Internship_Inventory`.`Internship`
-- -----
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`Internship` ;
CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`Internship` (
`InternshipId` VARCHAR(15) NOT NULL,
`StartDate` DATE NOT NULL,
`EndDate` DATE NOT NULL,
`Location` VARCHAR(20) NULL DEFAULT 'Not Decided',
`Description` VARCHAR(1000) NULL DEFAULT NULL,
`HoursPerWeek` INT(11) NOT NULL,
`NoOfOpenings` INT(11) NOT NULL,
`Title` VARCHAR(45) NOT NULL,
`PaidYorN` VARCHAR(15) NOT NULL,
`CompanyID` VARCHAR(15) NOT NULL,
```

```
PRIMARY KEY (`InternshipId`),  
  
INDEX `CompanyId_idx` (`CompanyId` ASC),  
  
UNIQUE INDEX `InternshipId_UNIQUE` (`InternshipId` ASC),  
  
INDEX `title` (`Title` ASC),  
  
CONSTRAINT `Internship_Company_CompanyID`  
  
FOREIGN KEY (`CompanyId`)  
  
REFERENCES `Internship_Inventory`.`Company` (`CompanyId`)  
  
ON DELETE NO ACTION  
  
ON UPDATE NO ACTION)  
  
ENGINE = InnoDB;
```

```
-- -----
```

```
-- Table `Internship_Inventory`.`Application`
```

```
-- -----
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`Application` ;  
  
CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`Application` (  
  
  `ApplicationID` VARCHAR(15) NOT NULL,  
  
  `HireOrNot` int NOT NULL DEFAULT 3,  
  
  `StudentID` VARCHAR(15) NOT NULL,  
  
  `InternshipID` VARCHAR(15) NOT NULL,  
  
  PRIMARY KEY (`ApplicationID`),  
  
  UNIQUE INDEX `ApplicationID_UNIQUE` (`ApplicationID` ASC),  
  
  INDEX `studentNo_idx` (`StudentID` ASC),  
  
  INDEX `InternshipIDs_idx` (`InternshipID` ASC),
```

```

CONSTRAINT `login_enum_HireOrNot`
  FOREIGN KEY (`HireOrNot`)
    REFERENCES `internship_inventory`.`enum` (`enumID`),
CONSTRAINT `Application_Student_studentId`
  FOREIGN KEY (`StudentID`)
    REFERENCES `Internship_Inventory`.`Student` (`StudentID`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `Application_Internship_InternshipId`
  FOREIGN KEY (`InternshipID`)
    REFERENCES `Internship_Inventory`.`Internship` (`InternshipId`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- Table `Internship_Inventory`.`Supervisor`

```

```

DROP TABLE IF EXISTS `Internship_Inventory`.`Supervisor` ;
CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`Supervisor` (
  `SupervisorID` VARCHAR(15) NOT NULL,
  `Name` VARCHAR(45) NOT NULL,
  `Email` VARCHAR(45) NOT NULL,
  `WorkPhone` VARCHAR(15) NULL,

```

```
`OfficeAddress` VARCHAR(90) NULL,  
`CompanyID` VARCHAR(90) NOT NULL,  
PRIMARY KEY (`SupervisorID`),  
INDEX `CompanyID_idx` (`CompanyID` ASC),  
CONSTRAINT `Company_ID`  
    FOREIGN KEY (`CompanyID`)  
    REFERENCES `Internship_Inventory`.`Company` (`CompanyId`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----
```

```
-- Table `Internship_Inventory`.`Placement`
```

```
-- -----
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`Placement` ;  
CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`Placement` (  
    `PlacementID` VARCHAR(15) NOT NULL,  
    `SupervisorID` VARCHAR(15) NOT NULL DEFAULT 'ToAssign',  
    `Notes` VARCHAR(45) NULL,  
    `StudentEvaluation` VARCHAR(255) DEFAULT 'TBA',  
    `SupervisorEvaluation` VARCHAR(255) DEFAULT 'TBA',  
    PRIMARY KEY (`PlacementID`),  
    INDEX `SupervisorID_idx` (`SupervisorID` ASC),  
    CONSTRAINT `Placement_Application_PlacementID`
```

```
FOREIGN KEY (`PlacementID`)
REFERENCES `Internship_Inventory`.`Application` (`ApplicationID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `Placement_Supervisor_Supervisor_ID`

FOREIGN KEY (`SupervisorID`)

REFERENCES `Internship_Inventory`.`Supervisor` (`SupervisorID`)

ON DELETE NO ACTION

ON UPDATE CASCADE)

ENGINE = InnoDB;
```

```
-- -----
```

```
-- Table `Internship_Inventory`.`Paperwork`
```

```
-- -----
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`Paperwork` ;

CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`Paperwork` (

`PlacementId` VARCHAR(15) NOT NULL,

`StudentEvaluation` VARCHAR(45) DEFAULT 'Generate from view StudentEvaluation',

`SupervisorEvaluation` VARCHAR(45) DEFAULT 'Generate from view SupervisorEvaluation',

`CompanyInfo` VARCHAR(45) DEFAULT 'Generate from view CompanyInfo',

`StudentInfo` VARCHAR(45) DEFAULT 'Generate from view StudentInfo',

PRIMARY KEY (`PlacementId`),

UNIQUE INDEX `PlacementId_UNIQUE` (`PlacementId` ASC),

CONSTRAINT `Placement_IDx`
```

```
FOREIGN KEY (`PlacementId`)  
REFERENCES `Internship_Inventory`.`Placement` (`PlacementID`)  
  
ON DELETE CASCADE  
  
ON UPDATE CASCADE)  
  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `Internship_Inventory`.`PaidIntern`  
-- -----
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`PaidIntern` ;  
  
CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`PaidIntern` (  
  `InternshipID` VARCHAR(15) NOT NULL,  
  `TotalSalary` DOUBLE NULL,  
  `WorkType` VARCHAR(45) NULL,  
  PRIMARY KEY (`InternshipID`),  
  UNIQUE INDEX `InternshipID_UNIQUE` (`InternshipID` ASC),  
  CONSTRAINT `InternshipIDx`  
    FOREIGN KEY (`InternshipID`)  
    REFERENCES `Internship_Inventory`.`Internship` (`InternshipId`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)  
  
ENGINE = InnoDB;
```

```
-- Table `Internship_Inventory`.`UnpaidIntern`  
-----  
  
DROP TABLE IF EXISTS `Internship_Inventory`.`UnpaidIntern` ;  
  
CREATE TABLE IF NOT EXISTS `Internship_Inventory`.`UnpaidIntern` (  
  `InternshipID` VARCHAR(15) NOT NULL,  
  `AcademicCredit` VARCHAR(45),  
  PRIMARY KEY (`InternshipID`),  
  UNIQUE INDEX `InternshipID_UNIQUE` (`InternshipID` ASC),  
  CONSTRAINT `Internship_IDx`  
    FOREIGN KEY (`InternshipID`)  
      REFERENCES `Internship_Inventory`.`Internship` (`InternshipId`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE)  
  
ENGINE = InnoDB;
```

```
-- Table `Internship_Inventory`.`enum`  
-----  
  
DROP TABLE IF EXISTS `internship_inventory`.`enum` ;  
  
CREATE TABLE IF NOT EXISTS `internship_inventory`.`enum` (  
  `enumId` int NOT NULL,  
  `content` VARCHAR(100) NOT NULL,  
  PRIMARY KEY (`enumId`))  
  
ENGINE = InnoDB;
```

-- Data for table `internship_inventory`.`enum`

-- Here need to insert first incase some value will be use later

START TRANSACTION;

USE `internship_inventory`;

for enum 'HireOrNot' in Table Application

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('10', 'No');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('11', 'Yes');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('12', 'Pending');

for enum 'access_rights' in Table login

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('14', 'ADMIN');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('15', 'USER');

for enum 'security_question' in Table login

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('17', 'What is your first movie?');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('18', 'What is your first pet\'s name?');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('19', 'In what year was your father born?');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('20', 'Where were you New Year\'s 2000?');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('21', 'Who is your childhood hero? ');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('22', 'What is the make and model of your first car?');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('23', 'What is your favorite sport in high school?');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('24', 'What school did you attend for sixth grade?');

INSERT INTO `internship_inventory`.`enum` (`enumID`, `content`) VALUES ('25', 'What is the name of your favorite childhood friend?');

COMMIT;


```
-- -----  
-- procedure computeCredits
```

```
-- -----  
  
USE `Internship_Inventory`;
```

```
DROP procedure IF EXISTS `Internship_Inventory`.`computeCredits`;
```

```
DELIMITER $$
```

```
USE `Internship_Inventory`$$
```

```
CREATE PROCEDURE `computeCredits` ()
```

```
BEGIN
```

```
    DECLARE done INT DEFAULT FALSE;
```

```
    DECLARE stdt, endt DATE;
```

```
    DECLARE hours INT;
```

```
    DECLARE id varchar(15);
```

```
    DECLARE cur1 CURSOR FOR SELECT
```

```
        INTERNSHIPID,STARTDATE,ENDDATE,HOURSPERWEEK
```

```
        FROM INTERNSHIP
```

```
    ;
```

```
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
```

```
    OPEN cur1;
```

```
read_loop: LOOP
```

```
    FETCH cur1 INTO id, stdt, endt, hours;
```

```
    IF done THEN
```

```
        LEAVE read_loop;
```

```

        END IF;

        UPDATE unpaidIntern SET AcademicCredit = hours*(DATEDIFF(endt,stdt)/7)*0.13

        WHERE internshipid=id;

    END LOOP;

    CLOSE cur1;

END$$

DELIMITER ;

```

```

-----

```

```

-- procedure dropdownlist

```

```

-----

```

```

-----

```

```

-- procedure dropdownlist1 onebyone (loop on web)

```

```

-----

```

```

USE `Internship_Inventory`;

```

```

DROP procedure IF EXISTS `Internship_Inventory`.`dropdownlist1`;

```

```

DELIMITER $$

```

```

USE `Internship_Inventory`$$

```

```

CREATE PROCEDURE `dropdownlist1` (IN param1 INT, OUT param2 varchar(45))

```

```

BEGIN

```

```

    select content into param2 from enum where enumId=param1;

```

```

END$$

```

```

DELIMITER ;

```

```

-----
-- procedure dropdownlist2 return multiple rows
-----

USE `Internship_Inventory`;

DROP procedure IF EXISTS `Internship_Inventory`.`dropdownlist2`;

DELIMITER $$

USE `Internship_Inventory`$$

CREATE PROCEDURE `dropdownlist2` (IN paramL INT, IN paramR INT)

BEGIN

    select content from enum where enumId<=paramR and enumId>=paramL;

END$$

DELIMITER ;

-----

-- trigger
-----

DELIMITER $$

USE `Internship_Inventory`$$

DROP TRIGGER IF EXISTS `Internship_Inventory`.`Application_AFTER_UPDATE` $$

USE `Internship_Inventory`$$

```

```
CREATE DEFINER = CURRENT_USER TRIGGER `Internship_Inventory`.`Application_AFTER_UPDATE` AFTER UPDATE ON `Application` FOR EACH ROW
BEGIN
if `old`.`hireornot` = 11 THEN
insert into paperwork (placementid) values (`placementid`);
end if;
END$$
```

```
USE `Internship_Inventory`$$
```

```
DROP TRIGGER IF EXISTS `Internship_Inventory`.`Placement_AFTER_UPDATE` $$
```

```
USE `Internship_Inventory`$$
```

```
CREATE DEFINER = CURRENT_USER TRIGGER `Internship_Inventory`.`Placement_AFTER_UPDATE` AFTER UPDATE ON `Placement` FOR EACH ROW
BEGIN
insert into paperwork (placementid) values (old.placementid);
END$$
```

```
-- -----
-- Trigger - Automatically add applicationID
-- when insert with format
-- -----
```

```
DELIMITER ;
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`application_seq` ;
```

```
CREATE TABLE application_seq
```

```
(
```

```
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY
```

```
);
```

```
DELIMITER $$
```

```
USE `Internship_Inventory`$$
```

```
DROP TRIGGER IF EXISTS `Internship_Inventory`.`tg_applicationId_insert` $$
```

```
CREATE TRIGGER tg_applicationId_insert
```

```
BEFORE INSERT ON application
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO application_seq VALUES (NULL);
```

```
SET NEW.applicationId = CONCAT('A',SUBSTRING(NEW.InterNshipID, 2, length(NEW.InterNshipID)-1),LAST_INSERT_ID());
```

```
END$$
```

```
-- -----
```

```
-- Trigger - Automatically insert paperwork
```

```
-- after insert placement
```

```
-- -----
```

```
USE `Internship_Inventory`$$
```

```
DROP TRIGGER IF EXISTS `Internship_Inventory`.`PAPERWORK_INSERT_PLACEMENT` $$
```

```
USE `Internship_Inventory`$$
```

```
CREATE DEFINER = CURRENT_USER TRIGGER `Internship_Inventory`.`PAPERWORK_INSERT_PLACEMENT`
```

```
AFTER INSERT ON `placement` FOR EACH ROW
```

BEGIN

insert into paperwork (placementid) values (new.placementid);

END\$\$

DELIMITER ;

/*

-- View-1 `Internship_Inventory`.`InternshipStudent`

DROP VIEW IF EXISTS `Internship_Inventory`.`InternshipStudent` ;

DROP TABLE IF EXISTS `Internship_Inventory`.`InternshipStudent`;

USE `Internship_Inventory`;

CREATE OR REPLACE VIEW `InternshipStudent` AS

select INTE.internshipID, INTE.TITLE, INTE.DESCRPTION, CO.COMPANYNAME, INTE.NOOFOPENINGS,CO.INDUSTRY

FROM COMPANY AS CO INNER JOIN INTERNSHIP AS INTE ON INTE.COMPANYID

=CO.COMPANYID;

*/

-- View-1 alternative

-- View-1 `Internship_Inventory`.`internshipDetails`

```

DROP VIEW IF EXISTS `Internship_Inventory`.`internshipDetails` ;

DROP TABLE IF EXISTS `Internship_Inventory`.`internshipDetails`;

USE `Internship_Inventory`;

CREATE OR REPLACE VIEW `internshipDetails` AS

select distinct INTE.internshipID, INTE.TITLE, INTE.DESCRPTION, skills.skillDescription, INTE.PAIDYORN, INTE.NOOFOPENINGS,CO.COMPANYNAME, CO.INDUSTRY
FROM COMPANY AS CO, INTERNSHIP AS INTE, skills, internshipskill
where INTE.COMPANYID=CO.COMPANYID and INTE.internshipID = internshipskill.InternshipID and internshipskill.SkillID=skills.SkillID;

```

```

-- -----
-- View-2 `Internship_Inventory`.`FallPlacement`
-- All placements for a particular time frame-Fall

```

```

-- -----

DROP VIEW IF EXISTS `Internship_Inventory`.`FallPlacement` ;

DROP TABLE IF EXISTS `Internship_Inventory`.`FallPlacement`;

USE `Internship_Inventory`;

CREATE OR REPLACE VIEW `FallPlacement` AS

SELECT distinct placement.placementID, person.name as studentName, company.CompanyName,
supervisor.name as supervisorName, application.internshipId, notes
FROM Placement, application, Person, supervisor, internship, company
WHERE PlacementID IN (SELECT ApplicationID FROM Application
WHERE application.StudentID IN (SELECT StudentID FROM student WHERE semester='Fall'))
and placementID = application.applicationID and application.studentID = person.ID
and application.internshipID = internship.internshipID and internship.companyID = company.companyID
and placement.supervisorID = supervisor.SupervisorID

```

;

-- View-3 `Internship_Inventory`.`StudentInfo`

-- All student information

DROP VIEW IF EXISTS `Internship_Inventory`.`StudentInfo` ;

DROP TABLE IF EXISTS `Internship_Inventory`.`StudentInfo`;

USE `Internship_Inventory`;

CREATE OR REPLACE VIEW `StudentInfo` AS

SELECT s.studentID, p.Name, p.cell, p.email, p.address, p.ssn, s.Semester, s.Major, s.Minor, s.GraduationYear, s.EnrollmentYear, s.GPA

FROM person p

RIGHT JOIN student s on p.ID=s.studentID;

-- View-4 `Internship_Inventory`.`CompanyInfo`

-- All company information including internship they provide

DROP VIEW IF EXISTS `Internship_Inventory`.`CompanyInfo` ;

DROP TABLE IF EXISTS `Internship_Inventory`.`CompanyInfo`;

USE `Internship_Inventory`;

CREATE OR REPLACE VIEW `CompanyInfo` AS

SELECT c.companyid, c.companyname, c.address, c.cell, c.description, c.email, industry, i.title, internshipid, noofopenings

FROM company c


```
LEFT JOIN internship i ON c.companyid=i.companyid;
```

```
-- -----
```

```
-- View-5 `Internship_Inventory`.`SupervisorEvaluation`
```

```
-- Supervisor's evaluation for company to review
```

```
-- -----
```

```
DROP VIEW IF EXISTS `Internship_Inventory`.`SupervisorEvaluation` ;
```

```
DROP TABLE IF EXISTS `Internship_Inventory`.`SupervisorEvaluation`;
```

```
USE `Internship_Inventory`;
```

```
CREATE OR REPLACE VIEW `SupervisorEvaluation` AS
```

```
SELECT placement.placementid, person.name as 'student_name', internship.title, semester, companyname, supervisor.supervisorid, supervisor.name as  
'supervisor_name', Placement.SupervisorEvaluation
```

```
FROM placement, person, internship, company, supervisor, application, student
```

```
WHERE placement.SupervisorEvaluation IS NOT NULL AND placement.placementid=application.applicationID AND application.studentID=student.studentID AND  
student.studentID=person.ID AND
```

```
application.internshipID=internship.internshipID AND internship.companyID=company.companyID AND
```

```
placement.supervisorID=supervisor.supervisorID
```

```
GROUP BY companyname;
```

```
-- -----
```

```
-- View-6 `Internship_Inventory`.`StudentEvaluation`
```

```
-- Student's evaluation on their completed internship
```

```
-- -----
```

```
DROP VIEW IF EXISTS `Internship_Inventory`.`StudentEvaluation` ;  
DROP TABLE IF EXISTS `Internship_Inventory`.`StudentEvaluation`;  
USE `Internship_Inventory`;  
CREATE OR REPLACE VIEW `StudentEvaluation` AS  
SELECT companyname, title, semester, person.name, studentevaluation  
FROM company, internship, student, person, placement,application  
WHERE placement.studentevaluation IS NOT NULL AND placement.placementID=application.applicationID AND  
application.studentID=student.studentID AND student.studentID=person.ID AND  
application.internshipID=internship.internshipID AND internship.companyID=company.companyID  
GROUP BY companyname;
```

```
-- INSERT DATA
```

```
-- -----  
-- Data for table `internship_inventory`.`login`  
-- -----
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`login` (`LoginId`, `Password`, `Security_Question`, `Security_Answer`, `Access_Rights`) VALUES ('10001', '123456', 17, 'Jane', 14);
```

```
INSERT INTO `internship_inventory`.`login` (`LoginId`, `Password`, `Security_Question`, `Security_Answer`, `Access_Rights`) VALUES ('10002', '234567', 18, 'Katty', 15);
```

```
INSERT INTO `internship_inventory`.`login` (`LoginId`, `Password`, `Security_Question`, `Security_Answer`, `Access_Rights`) VALUES ('10003', '456789', 19, '1900', 15);
```

```
INSERT INTO `internship_inventory`.`login` (`LoginId`, `Password`, `Security_Question`, `Security_Answer`, `Access_Rights`) VALUES ('10004', '987655', 20, 'New York', 15);
```

```
COMMIT;
```

```
-- Data for table `internship_inventory`.`person`
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`person` (`ID`, `Name`, `Cell`, `Email`, `Address`, `SSN`, `LoginID`) VALUES ('10000', 'John Lee', '(251) 578-9442', 'jlee@gmail.com', '111 Deerwood Road, San Ramon, CA 94583', '511-22-1993', '10000');
```

```
INSERT INTO `internship_inventory`.`person` (`ID`, `Name`, `Cell`, `Email`, `Address`, `SSN`, `LoginID`) VALUES ('10001', 'Josefina Douglas', '(251) 546-9442', 'mthurn@live.com', '20 Maple Avenue, San Pedro, CA 90731', '520-22-0066', '10001');
```

```
INSERT INTO `internship_inventory`.`person` (`ID`, `Name`, `Cell`, `Email`, `Address`, `SSN`, `LoginID`) VALUES ('10002', 'Johnnie Green', '(125) 546-4478', 'rgarcia@optonline.net', '601 Sherwood Ave. San Bernardino, CA 92404', '527-42-8884', '10002');
```

```
INSERT INTO `internship_inventory`.`person` (`ID`, `Name`, `Cell`, `Email`, `Address`, `SSN`, `LoginID`) VALUES ('10003', 'Angelica Lindsey', '(226) 906-2721', 'webdragon@comcast.net', '7246 W. Windsor Dr. Carmichael, CA 95608', '525-47-9158', '10003');
```

```
INSERT INTO `internship_inventory`.`person` (`ID`, `Name`, `Cell`, `Email`, `Address`, `SSN`, `LoginID`) VALUES ('10004', 'Shelia Curtis', '(671) 925-1352', 'crandall@sbcglobal.net', '241 Indian Spring St. Pittsburg, CA 94565', '527-80-4977', '10004');
```

```
INSERT INTO `internship_inventory`.`person` (`ID`, `Name`, `Cell`, `Email`, `Address`, `SSN`, `LoginID`) VALUES ('10005', 'Daniel Brown', '(732) 925-1352', 'dbrown@sbcglobal.net', '3336 Indian Spring New Brunswick 08933', '511-80-4934', '10005');
```

```
INSERT INTO `internship_inventory`.`person` (`ID`, `Name`, `Cell`, `Email`, `Address`, `SSN`, `LoginID`) VALUES ('10006', 'Christ Kirby', '(671) 925-1352', 'ckirby@yahoo.com', '241 Indian Spring St. Pittsburg, CA 94565', '522-80-4923', '10006');
```

```
INSERT INTO `internship_inventory`.`person` (`ID`, `Name`, `Cell`, `Email`, `Address`, `SSN`, `LoginID`) VALUES ('10007', 'Stephen Young', '(212) 925-1352', 'syoung@sbcglobal.net', '70 Washington Square South New York, NY 10003', '533-80-4912', '10007');
```

```
COMMIT;
```

```
-- Data for table `internship_inventory`.`employee`
```

START TRANSACTION;

USE `internship_inventory`;

INSERT INTO `internship_inventory`.`employee` (`EmployeeID`, `OfficePhone`, `OfficeAdd`) VALUES ('10000', '5866683878', '2600 Clifton Ave., Cincinnati, OH 45221');

INSERT INTO `internship_inventory`.`employee` (`EmployeeID`, `OfficePhone`, `OfficeAdd`) VALUES ('10001', '4564883874', '123 6th St. Melbourne, FL 32904');

INSERT INTO `internship_inventory`.`employee` (`EmployeeID`, `OfficePhone`, `OfficeAdd`) VALUES ('10002', '5866684389', '71 Pilgrim Avenue, Chevy Chase, MD 20815');

INSERT INTO `internship_inventory`.`employee` (`EmployeeID`, `OfficePhone`, `OfficeAdd`) VALUES ('10003', '8883158641', '70 Bowman St. South Windsor, CT 06074');

COMMIT;

-- Data for table `internship_inventory`.`faculty`

START TRANSACTION;

USE `internship_inventory`;

INSERT INTO `internship_inventory`.`faculty` (`FacID`, `Dept`, `Type`) VALUES ('10000', 'Math', 'Fulltime');

INSERT INTO `internship_inventory`.`faculty` (`FacID`, `Dept`, `Type`) VALUES ('10001', 'Math', 'Fulltime');

INSERT INTO `internship_inventory`.`faculty` (`FacID`, `Dept`, `Type`) VALUES ('10003', 'Art', 'Parttime');

COMMIT;

-- Data for table `internship_inventory`.`administrator`

START TRANSACTION;

USE `internship_inventory`;

```
INSERT INTO `internship_inventory`.`administrator` (`AdminID`, `Shift`) VALUES ('10002', 'Yes');
```

```
COMMIT;
```

```
-- -----
```

```
-- Data for table `internship_inventory`.`student`
```

```
-- -----
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`student` (`StudentID`, `Semester`, `Major`, `Minor`, `GraduationYear`, `EnrollmentYear`, `GPA`) VALUES ('10004', 'Spring', 'Math', NULL, 2020, 2016, 3.5);
```

```
INSERT INTO `internship_inventory`.`student` (`StudentID`, `Semester`, `Major`, `Minor`, `GraduationYear`, `EnrollmentYear`, `GPA`) VALUES ('10005', 'Fall', 'Art', NULL, 2018, 2016, 3.2);
```

```
INSERT INTO `internship_inventory`.`student` (`StudentID`, `Semester`, `Major`, `Minor`, `GraduationYear`, `EnrollmentYear`, `GPA`) VALUES ('10006', 'Spring', 'CS', NULL, 2017, 2015, 3.7);
```

```
INSERT INTO `internship_inventory`.`student` (`StudentID`, `Semester`, `Major`, `Minor`, `GraduationYear`, `EnrollmentYear`, `GPA`) VALUES ('10007', 'Fall', 'Finance', NULL, 2022, 2018, 3.0);
```

```
COMMIT;
```

```
-- -----
```

```
-- Data for table `internship_inventory`.`company`
```

```
-- -----
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`company` (`CompanyID`, `CompanyName`, `Address`, `Cell`, `Description`, `Industry`, `Email`) VALUES ('C1', 'Walmart', '702 Sw 8th St, Bentonville, AR', '(479) 273-4000', 'Wal-Mart Stores is an irresistible (or at least unavoidable) retail force that has yet to meet any immovable objects. The world\'s largest company by revenue and bigger than Europe\'s Carrefour, Metro AG, and Tesco combined, Walmart is the world\'s #1 retailer with more than 2.3 million employees. In the US, Wal-Mart operates more than 5,300 stores, including about 4,570 Wal-Mart stores and 655 Sam\'s Club warehouses, and a growing number of smaller format stores. The company\'s faster growing international division (26% of sales) numbers more than 6,100 locations; Wal-Mart is
```

the #1 retailer in Canada and Mexico and has operations in Asia (where it owns a 95% stake in Japanese retailer SEIYU), Africa, Europe, and Latin America.',
'Retail', 'www.walmart.com');

INSERT INTO `internship_inventory`.`company` (`CompanyID`, `CompanyName`, `Address`, `Cell`, `Description`, `Industry`, `Email`) VALUES ('C2', 'Exxon Mobil Corporation', '4045 Scenic Hwy, Baton Rouge, LA', '(985) 359-8519', 'Some telephone numbers on the Hoover’s site may be on a country’s do not call or do not contact list including, but not limited to, the United Kingdom’s CTPS or TPS registers. It is a legal requirement that companies do not make sales or marketing calls to registered numbers. These are central opt out registers whereby corporate subscribers and individuals can register their preference not to receive unsolicited sales and marketing telephone calls. By using the information provided on the Hoover’s sites, as the direct marketer you represent and warrant that you will use such information in compliance with all applicable local, state, national or international laws and regulations, including any local do not call registers or marketing regulations, and agree to defend, indemnify and hold harmless Dun & Bradstreet and each of its affiliates in the event your use violates such laws and regulations.', 'Petroleum refining', 'www.exxonmobil.com');

INSERT INTO `internship_inventory`.`company` (`CompanyID`, `CompanyName`, `Address`, `Cell`, `Description`, `Industry`, `Email`) VALUES ('C3', 'TOYOTA MOTOR ', '1, Toyotacho, Toyota, Aichi,Japan', '+81-565282121', 'Toyota Motor, among the world’s largest automotive manufacturers by revenue, designs and manufactures a diverse product line-up that ranges from subcompacts to luxury and sports vehicles to SUVs, trucks, minivans, and buses. Its vehicles are produced either with combustion or hybrid engines, as with the Prius. Toyota’s subsidiaries also manufacture vehicles: Daihatsu Motor produces mini-vehicles, while Hino Motors produces trucks and buses. Additionally, Toyota makes automotive parts for its own use and for sale to others. Popular models include the Camry, Corolla, Land Cruiser, and luxury Lexus line, as well as the Tundra truck.', 'automotive manufacturers', 'www.toyota.com');

INSERT INTO `internship_inventory`.`company` (`CompanyID`, `CompanyName`, `Address`, `Cell`, `Description`, `Industry`, `Email`) VALUES ('C4', 'Apple Inc.', '1 Infinite Loop, Cupertino, CA', '(408) 996-1010', 'Ask Siri to name the most successful company in the world and it might respond: Apple. In terms of profit, revenue, market capitalization, and consumer cachet, it certainly ranks right up there. The iPhone, in its ninth year and seventh generation, has been the company’s golden goose, although the aging product may be losing a little of its luster. In addition to the iPhone, other familiar Apple products and services include MacBook computers and iPad tablets, as well as iTunes, the App store, and Apple Music. Primarily a consumer-oriented company, Apple has inked alliances with IBM and Cisco Systems to deepen its penetration of the enterprise market. About 60% of revenue comes from outside the Americas.', 'technology', 'www.apple.com');

COMMIT;

-- Data for table `internship_inventory`.`internship`

START TRANSACTION;

USE `internship_inventory`;

```
INSERT INTO `internship_inventory`.`internship` (`InternshipId`, `StartDate`, `EndDate`, `Location`, `Description`, `HoursPerWeek`, `NoOfOpenings`, `Title`, `PaidYorN`, `CompanyID`) VALUES ('I101', '2017-05-01', '2017-07-01', 'Charlotte', 'The successful candidate will turn data into information, information into insight and insight into business decisions. Data analyst responsibilities include conducting full lifecycle analysis to include requirements, activities and design. Data analysts will develop analysis and reporting capabilities. They will also monitor performance and quality control plans to identify improvements.', 20, 2, 'Data Analytics', 'Y', 'C1');
```

```
INSERT INTO `internship_inventory`.`internship` (`InternshipId`, `StartDate`, `EndDate`, `Location`, `Description`, `HoursPerWeek`, `NoOfOpenings`, `Title`, `PaidYorN`, `CompanyID`) VALUES ('I201', '2017-05-01', '2017-10-01', 'New York', 'Marketing Assistant responsibilities include assisting in organizing campaigns and developing marketing strategies. This work will be a critical factor for the smooth operation of the Marketing department and the attainment of its goals, as well as for the long-term growth of the company.', 40, 3, 'Marketing', 'Y', 'C2');
```

```
INSERT INTO `internship_inventory`.`internship` (`InternshipId`, `StartDate`, `EndDate`, `Location`, `Description`, `HoursPerWeek`, `NoOfOpenings`, `Title`, `PaidYorN`, `CompanyID`) VALUES ('I301', '2018-07-01', '2019-07-01', 'Chicago', 'Java developer roles and responsibilities include managing Java/Java EE application development while providing expertise in the full software development lifecycle, from concept and design to testing.', 10, 1, 'Developer', 'N', 'C3');
```

```
INSERT INTO `internship_inventory`.`internship` (`InternshipId`, `StartDate`, `EndDate`, `Location`, `Description`, `HoursPerWeek`, `NoOfOpenings`, `Title`, `PaidYorN`, `CompanyID`) VALUES ('I401', '2018-07-01', '2018-10-01', 'San Francisco', 'Sales assistant duties include selling, restocking and merchandising. The goal is to provide high class customer service and to increase company's growth and revenue through sales maximisation.', 30, 5, 'Sales', 'N', 'C4');
```

```
COMMIT;
```

```
-- -----  
-- Data for table `internship_inventory`.`application`  
-- -----
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`application` (`ApplicationID`, `HireOrNot`, `StudentID`, `InternshipID`) VALUES ('A1011', 10, '10004', 'I101');
```

```
INSERT INTO `internship_inventory`.`application` (`ApplicationID`, `HireOrNot`, `StudentID`, `InternshipID`) VALUES ('A2011', 11, '10005', 'I201');
```

```
INSERT INTO `internship_inventory`.`application` (`ApplicationID`, `HireOrNot`, `StudentID`, `InternshipID`) VALUES ('A3011', 11, '10007', 'I301');
```

```
INSERT INTO `internship_inventory`.`application` (`ApplicationID`, `HireOrNot`, `StudentID`, `InternshipID`) VALUES ('A4011', 12, '10006', 'I401');
```

```
INSERT INTO `internship_inventory`.`application` (`HireOrNot`, `StudentID`, `InternshipID`) VALUES (12, '10008', 'I402');
```

```
COMMIT;
```

```
-- Data for table `internship_inventory`.`skills`
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`skills` (`SkillID`, `SkillDescription`) VALUES ('S10001', 'Java');
```

```
INSERT INTO `internship_inventory`.`skills` (`SkillID`, `SkillDescription`) VALUES ('S10002', 'SQL');
```

```
INSERT INTO `internship_inventory`.`skills` (`SkillID`, `SkillDescription`) VALUES ('S10003', 'Python');
```

```
INSERT INTO `internship_inventory`.`skills` (`SkillID`, `SkillDescription`) VALUES ('S10004', 'Excel');
```

```
COMMIT;
```

```
-- Data for table `internship_inventory`.`internshipskill`
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`internshipskill` (`SkillID`, `InternshipID`) VALUES ('S10001', 'I101');
```

```
INSERT INTO `internship_inventory`.`internshipskill` (`SkillID`, `InternshipID`) VALUES ('S10002', 'I201');
```

```
INSERT INTO `internship_inventory`.`internshipskill` (`SkillID`, `InternshipID`) VALUES ('S10001', 'I201');
```

```
INSERT INTO `internship_inventory`.`internshipskill` (`SkillID`, `InternshipID`) VALUES ('S10003', 'I301');
```

```
INSERT INTO `internship_inventory`.`internshipskill` (`SkillID`, `InternshipID`) VALUES ('S10001', 'I401');
```

```
INSERT INTO `internship_inventory`.`internshipskill` (`SkillID`, `InternshipID`) VALUES ('S10004', 'I401');
```



```
COMMIT;
```

```
-- Data for table `internship_inventory`.`paidintern`
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`paidintern` (`InternshipID`, `TotalSalary`, `WorkType`) VALUES ('I101', 20000.0, 'Remote');
```

```
INSERT INTO `internship_inventory`.`paidintern` (`InternshipID`, `TotalSalary`, `WorkType`) VALUES ('I201', 30000.0, 'Onsite');
```

```
COMMIT;
```

```
-- Data for table `internship_inventory`.`unpaidintern`
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`unpaidintern` (`InternshipID`, `AcademicCredit`) VALUES ('I301', null);
```

```
INSERT INTO `internship_inventory`.`unpaidintern` (`InternshipID`, `AcademicCredit`) VALUES ('I401', null);
```

```
call computeCredits();
```

```
COMMIT;
```

```
-- Data for table `internship_inventory`.`supervisor`
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`supervisor` (`SupervisorID`, `Name`, `Email`, `WorkPhone`, `OfficeAddress`, `CompanyID`) VALUES ('S10001', 'Myriam Oneill', 'mxiao@yahoo.com', '(251) 546-9442', '71 Pilgrim Avenue, Chevy Chase, MD 20815 ', '10002');
```

```
INSERT INTO `internship_inventory`.`supervisor` (`SupervisorID`, `Name`, `Email`, `WorkPhone`, `OfficeAddress`, `CompanyID`) VALUES ('S10002', 'Johanne Rawlings', 'dogdude@comcast.net', '(226) 906-2721', '44 Shirley Ave. West Chicago, IL 60185', '10003');
```

```
COMMIT;
```

```
-- Data for table `internship_inventory`.`placement`
```

```
START TRANSACTION;
```

```
USE `internship_inventory`;
```

```
INSERT INTO `internship_inventory`.`placement` (`PlacementID`, `SupervisorID`, `Notes`, `StudentEvaluation`, `SupervisorEvaluation`) VALUES ('A1011', 'S10001', 'Pending', 'Pleasantworkplace', 'Creative');
```

```
INSERT INTO `internship_inventory`.`placement` (`PlacementID`, `SupervisorID`, `Notes`, `StudentEvaluation`, `SupervisorEvaluation`) VALUES ('A2011', 'S10002', 'Pending', 'Sometimes work overtime but learn a lot of technical skills', 'Hardworking');
```

```
INSERT INTO `internship_inventory`.`placement` (`PlacementID`, `SupervisorID`, `Notes`, `StudentEvaluation`, `SupervisorEvaluation`) VALUES ('A2012', 'S10001', 'Pending', '', '');
```

```
INSERT INTO `internship_inventory`.`placement` (`PlacementID`, `SupervisorID`, `Notes`, `StudentEvaluation`, `SupervisorEvaluation`) VALUES ('A3013', 'S10002', 'Pending', '', '');
```

```
COMMIT;
```

```
SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

-- -----

-- Delete Data for table `internship_inventory`.`person`

-- -----

-- turn on CHECKS

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=1;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=1;

# SET SQL_SAFE_UPDATES = 1;
```

CONCLUSION

A relational database is a wonderful piece of equipment for storing large quantities of data efficiently. In this project we focused mainly on building the database model. These models can be implemented on any RDBMS and queried using the Structured Query Language. We created a database that a University can use for keeping track of the students internships. The users of the system are classified as Students, Supervisors and Employees who are further classified as Admin and Faculty.

In our case we have created the database according to the requirements that have been specified during the start of the project. This Internship Management System makes it easy for the University Management to keep track of all the students placement details instead of using paperwork for each and every student.

During our database management course we have learned about the basics of database design. This project gave us the opportunity to try our new skills in practice. While doing this project we also gained deeper understanding on database design and how it can be implemented in real life situations.

REFERENCES

1. <https://dev.mysql.com/doc/>
2. <https://stackoverflow.com/tags/jsp/info>
3. <https://spring.io/tools>
4. <https://www.journaldev.com/1854/java-web-application-tutorial-for-beginners>
5. <https://www.service-architecture.com/articles/database/index.html>
6. <https://www.tutorialspoint.com/dbms/>
7. [https://technet.microsoft.com/en-us/library/ms191436\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms191436(v=sql.105).aspx)
8. <https://docs.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql>
9. <https://www.w3schools.com/css/>