



A

CASE STUDY OF
INTERNSHIP MANAGEMENT
SYSTEM
IN

Relational Database Management System

Submitted to: Prof. Solanki

Team Information

Team Leader

S.No.	Name	Roll Number
1.	Parikshit Pareek	2010991649

Team Members

S.No.	Name	Roll Number
1.	Luckshay	2010991610
2.	Niket Chawla	2010991638
3.	Om Saini	2010991645
4.	Chirag Sharma	2010991784

Introduction: Origin of Proposal

Indian internship industry is one of the **fastest-growing** in the world. India is among the **highest** in the world in terms of young workforce availability. Moreover, India's foreign investment sector is experiencing exponential growth with companies shifting to India, taking place not just in major cities and metros, but also in **tier II and tier III cities**.

What is **shocking**, however, is how even after such projected rapid growth and development, interns all across the country mostly rely either on **memory** for managing their internship applications; or rely mostly on **non-RDBMS** software such as Microsoft excel.

Following any of the above-mentioned ways has the following disadvantages and challenges:

- i. It takes a huge amount of **time** to manage the complete system manually
- ii. A manual system does not help to provide a **quick view** of applications accepted, membership details, faculty data, etc.
- iii. There are always good chances of **errors** while handling this information manually
- iv. The storage and preservation of **large volumes** of data become difficult.

- v. Non - RDBMS applications not only **lack basic features** such as trigger activation but also have no option like **multi-view** due to which confidential information of students, faculty, companies and the entire database, in general, can be compromised.

In conclusion, there is a **dire need** for a product that can overcome all the above-stated challenges and limitations without making the complete process complex.

Objective: Statement of Purpose

Through the means of this case study, the aim is to **overcome** all the before-stated challenges and limitations by introducing a holistic and easy-to-use **internship management system**.

To introduce an embedded internship management system that will **track interns and companies** from the point of application through record keeping to the point of getting placed and **automate** most tasks involved in tracking the **application for internships in real-time**.

To inculcate a database management system that will keep a record of all the interns and companies and that are willing to be interns, and are wanting to take in the same, respectively through a membership system.

To introduce triggers that remind both the interns and companies some days before the tenure of their membership ends.

In conclusion, the **main objective** of the project is to provide storekeepers with a more **efficient, robust, and comprehensive** internship management system.

Abstract

One case study “**internship management system**” is presented. Input for this case study is taken from its informal specification to a **relational schema** using **entity-relationship modelling** and its translation to relational model, to database schema, to **implementation of the database**, to **interactive SQL querying** of the installed database (SQL/Oracle).

Acknowledgments

We would like to express our gratitude to all of those who made it possible to complete this thesis, in **particular** to my supervisor **Prof. Vikas Solanki**. We would also like to extend gratitude to our families and friends for their understanding and continuous support.

Technical Specifications

Software Requirement Specification:

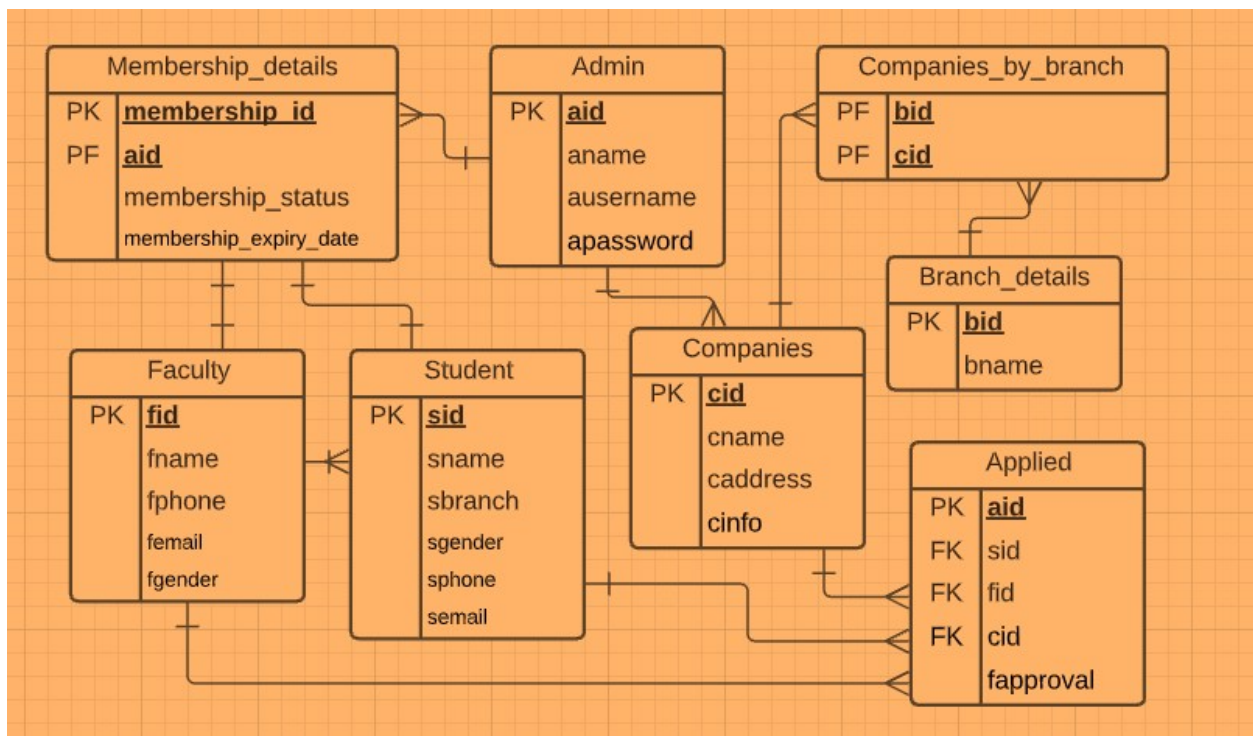
- i. Windows OS
- ii. RDBMS backend – SQL Server (via Oracle preferably)

Hardware Requirement Specification:

- i. 4 GB RAM
- ii. 120 GB Hard disk storage
- iii. A working computer with basic input-output peripherals

Logical Model

The ER Representation of the project is as followed:



The above diagram has four types of relationships:

- ❖ One to many
- ❖ Many to one
- ❖ One to one
- ❖ Many to many

- There will be multiple admins (for different departments) associated with multiple faculty members. So, there will be M:M relationship between them. To remove that a link is created in between them named membership_details.
- Student can apply to multiple companies so there is one to multiple relationship between them.
- Multiple faculty members can approve the applied request of students.
- Multiple companies will be there based on multiple branches. So, there is many to many relationships. To remove or split it into two one to many relationships a relation is created in between them named companies_by_branch.
- One admin can make record of multiple companies so it has one to many relationships.

Physical Model

Create Statements

1.

```
create table admin(  
aid number primary key,  
aname varchar2(50),  
ausername varchar2(50),  
apassword varchar2(50));
```

```
Connected to:  
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
  
SQL> create table admin(aid number primary key, aname varchar2(50), ausername varchar2(50), apassword varchar2(50));  
Table created.  
  
SQL>
```

2.

```
create table student(  
sid number primary key,  
sname varchar2(50),  
sbranch varchar2(10),  
sgender varchar2(10),  
sphone number,  
semail varchar2(50));
```

```
Connected to:  
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
  
SQL> create table admin(aid number primary key, aname varchar2(50), ausername varchar2(50), apassword varchar2(50));  
Table created.  
  
SQL> create table student(sid number primary key, sname varchar2(50), sbranch varchar2(10), sgender varchar2(10), sphone  
number, semail varchar2(50));  
Table created.  
  
SQL>
```

python3 change_ch10_files.py3.

```
create table companies(
cid number primary key,
cname varchar2(100),
caddress varchar2(1000),
cinfo varchar2(1000));
```

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> create table companies(cid number primary key, cname varchar2(100), caddress varchar2(1000), cinfo varchar2(1000));

Table created.

SQL>
```

4.

```
create table branch_details(
bid number primary key,
bname varchar2(10));
```

```
SQL> create table branch_details(bid number primary key, bname varchar2(10));

Table created.

SQL>
```

5.

```
create table faculty(
fid number primary key,
fname varchar2(50),
fphone number,
femail varchar2(50),
fgender varchar2(10));
```

```
SQL> create table faculty(fid number primary key, fname varchar2(50), fphone number, femail varchar2(50), fgender varchar2(10));

Table created.

SQL>
```

6.

```
create table membership_details(
membership_id number,
aid number,
membership_status varchar2(50),
membership_expiry_date date,
constraint pk_1 primary key(membership_id, aid),
constraint fk_1 foreign key(aid) references admin(aid));
```

```
SQL> create table membership_details(membership_id number, aid number, membership_status varchar2(50),membership_expiry_
date date,constraint pk_1 primary key(membership_id, aid), constraint fk_1 foreign key(aid) references admin(aid));
Table created.
SQL>
```

7.

```
create table companies_by_branch(
bid number,
cid number,
constraint pk_2 primary key(bid, cid),
constraint fk_2 foreign key(bid) references branch_details(bid),
constraint fk_3 foreign key(cid) references companies(cid));
```

```
SQL> create table companies_by_branch(bid number, cid number, constraint pk_2 primary key(bid, cid), constraint fk_2 for
eign key(bid) references branch_details(bid), constraint fk_3 foreign key(cid) references companies(cid));
Table created.
SQL>
```

8.

```
create table applied(
aid number primary key,
sid number,
fid number,
cid number,
fapproval varchar2(10),
constraint fk_4 foreign key(sid) references student(sid),
constraint fk_5 foreign key(fid) references faculty(fid),
constraint fk_6 foreign key(cid) references companies(cid));
```

```
SQL> create table applied(aid number primary key, sid number, fid number, cid number, fapproval varchar2(10), constraint
fk_4 foreign key(sid) references student(sid),constraint fk_5 foreign key(fid) references faculty(fid),constraint fk_6
foreign key(cid) references companies(cid));
Table created.
SQL>
```

Insert Statements

1. Admin Table

```
insert into admin values(1,'a1','a11','a1@1234');
insert into admin values(2,'a2','a21','a2@1234');
insert into admin values(3,'a3','a31','a3@1234');
insert into admin values(4,'a4','a41','a4@1234');
insert into admin values(5,'a5','a51','a5@1234');
```

```
SQL> insert into admin values(1,'a1','a11','a1@1234');
1 row created.
SQL> insert into admin values(2,'a2','a21','a2@1234');
1 row created.
SQL> insert into admin values(3,'a3','a31','a3@1234');
1 row created.
SQL> insert into admin values(4,'a4','a41','a4@1234');
1 row created.
SQL> insert into admin values(5,'a5','a51','a5@1234');
1 row created.
SQL>
```

```
SQL> select * from admin;
```

AID	ANAME	AUSERNAME	APASSWORD
1	a1	a11	a1@1234
2	a2	a21	a2@1234
3	a3	a31	a3@1234
4	a4	a41	a4@1234
5	a5	a51	a5@1234

```
SQL>
```

2. Student Table

```
insert into student
values(1,'s1','b1','m',8968189896,'mehraanaf@gmail.com');

insert into student
values(2,'s2','b1','m',6969696969,'mehengamessi@gmail.com');

insert into student
values(3,'s3','b2','f',9056723231,'saregamaluckshay@gmail.com');
```

```
insert into student
values (4,'s4','b2','f',9051723231,'promotenc@gmail.com');
```

```
insert into student
values (5,'s5','b2','f',9121723231,'marvellubber@gmail.com');
```

```
SQL> insert into student values(1,'s1','b1','m',9056723456,'s1@gmail.com');
1 row created.

SQL> insert into student values(2,'s2','b1','m',9056723451,'s2@gmail.com');
1 row created.

SQL> insert into student values(3,'s3','b2','f',9056723231,'s3@gmail.com');
1 row created.

SQL> insert into student values(4,'s4','b2','f',9051723231,'s4@gmail.com');
1 row created.

SQL> insert into student values(5,'s5','b2','f',9121723231,'s5@gmail.com');
1 row created.

SQL>
```

```
SQL> select * from student;
```

SID	SNAME	SBRANCH	SGENDER	SPHONE	SEMAIL
1	s1	b1	m	9056723456	s1@gmail.com
2	s2	b1	m	9056723451	s2@gmail.com
3	s3	b2	f	9056723231	s3@gmail.com
4	s4	b2	f	9051723231	s4@gmail.com
5	s5	b2	f	9121723231	s5@gmail.com

```
SQL>
```

3. Companies Table

```
insert into companies
values (1,'c1','c1_address','c1_private_limited_details');
```

```
insert into companies
values (2,'c2','c2_address','c2_private_limited_details');
```

```
insert into companies
values (3,'c3','c3_address','c3_private_limited_details');
```

```
insert into companies
```

```
values(4,'c4','c4_address','c4_private_limited_details');
```

```
insert into companies
```

```
values(5,'c5','c5_address','c5_private_limited_details');
```

```
SQL> insert into companies values(1,'c1','c1_address','c1_private_limited_details');
1 row created.

SQL> insert into companies values(2,'c2','c2_address','c2_private_limited_details');
1 row created.

SQL> insert into companies values(3,'c3','c3_address','c3_private_limited_details');
1 row created.

SQL> insert into companies values(4,'c4','c4_address','c4_private_limited_details');
1 row created.

SQL> insert into companies values(5,'c5','c5_address','c5_private_limited_details');
1 row created.

SQL>
```

4. Faculty Table

```
insert into faculty
```

```
values(1,'f1',9234567811,'f1@gmail.com','m');
```

```
insert into faculty
```

```
values(2,'f2',7234567811,'f2@gmail.com','f');
```

```
insert into faculty
```

```
values(3,'f3',7264567811,'f3@gmail.com','f');
```

```
insert into faculty
```

```
values(4,'f4',7114567811,'f4@gmail.com','m');
```

```
insert into faculty
```

```
values(5,'f5',7214567811,'f5@gmail.com','m');
```



```
SQL> insert into faculty values(1,'f1',9234567811,'f1@gmail.com','m');
1 row created.

SQL> insert into faculty values(2,'f2',7234567811,'f2@gmail.com','f');
1 row created.

SQL> insert into faculty values(3,'f3',7264567811,'f3@gmail.com','f');
1 row created.

SQL> insert into faculty values(4,'f4',7114567811,'f4@gmail.com','m');
1 row created.

SQL> insert into faculty values(5,'f5',7214567811,'f5@gmail.com','m');
1 row created.

SQL>
```

```
SQL> select * from faculty;
```

FID	FNAME	FPHONE	FEMAIL	FGENDER
1	f1	9234567811	f1@gmail.com	m
2	f2	7234567811	f2@gmail.com	f
3	f3	7264567811	f3@gmail.com	f
4	f4	7114567811	f4@gmail.com	m
5	f5	7214567811	f5@gmail.com	m

```
SQL>
```

5. Branch Details Table

```
insert into branch_details values(1,'b1');
insert into branch_details values(2,'b2');
insert into branch_details values(3,'b3');
insert into branch_details values(4,'b4');
```

```
SQL> insert into branch_details values(1,'b1');
1 row created.

SQL> insert into branch_details values(2,'b2');
1 row created.

SQL> insert into branch_details values(3,'b3');
1 row created.

SQL> insert into branch_details values(4,'b4');
1 row created.
```

```
SQL> select * from branch_details;

      BID BNAME
-----
      1 b1
      2 b2
      3 b3
      4 b4

SQL>
```

6. Companies by Branch Table

```
insert into companies_by_branch values(1,1);
insert into companies_by_branch values(1,2);
insert into companies_by_branch values(1,3);
insert into companies_by_branch values(2,1);
insert into companies_by_branch values(2,4);
```

```
SQL> insert into companies_by_branch values(1,1);
1 row created.

SQL> insert into companies_by_branch values(1,2);
1 row created.

SQL> insert into companies_by_branch values(1,3);
1 row created.

SQL> insert into companies_by_branch values(2,1);
1 row created.

SQL> insert into companies_by_branch values(2,4);
1 row created.

SQL>
```

```
SQL> select * from companies_by_branch;

      BID      CID
-----
        1        1
        1        2
        1        3
        2        1
        2        4

SQL>
```

7. Membership Details Table

```
insert into membership_details values(1,1,'active',to_date('13-07-2022','DD-MM-YYYY'));
```

```
insert into membership_details values(2,1,'active',to_date('21-07-2022','DD-MM-YYYY'));
```

```
insert into membership_details values(3,1,'active',to_date('25-10-2022','DD-MM-YYYY'));
```

```
insert into membership_details values(4,2,'active',to_date('25-10-2022','DD-MM-YYYY'));
```

```
insert into membership_details values(5,2,'active',to_date('25-10-2022','DD-MM-YYYY'));
```

```
SQL> insert into membership_details values(1,1,'active',to_date('13-07-2022','DD-MM-YYYY'));
1 row created.

SQL> insert into membership_details values(2,1,'active',to_date('21-07-2022','DD-MM-YYYY'));
1 row created.

SQL> insert into membership_details values(3,1,'active',to_date('25-10-2022','DD-MM-YYYY'));
1 row created.

SQL> insert into membership_details values(4,2,'active',to_date('25-10-2022','DD-MM-YYYY'));
1 row created.

SQL> insert into membership_details values(5,2,'active',to_date('25-10-2022','DD-MM-YYYY'));
1 row created.

SQL>
```

```
SQL> select * from membership_details;
```

MEMBERSHIP_ID	AID	MEMBERSHIP_STATUS	MEMBERSHI
1	1	active	13-JUL-22
2	1	active	21-JUL-22
3	1	active	25-OCT-22
4	2	active	25-OCT-22
5	2	active	25-OCT-22

```
SQL>
```

8. Applied Table

```
insert into branch_details values(1,'b1');
insert into branch_details values(2,'b2');
insert into branch_details values(3,'b3');
insert into branch_details values(4,'b4');
```

```
SQL> insert into applied values(1,1,1,1,'yes');
1 row created.

SQL> insert into applied values(2,2,1,1,'yes');
1 row created.

SQL> insert into applied values(3,2,1,2,'yes');
1 row created.

SQL> insert into applied values(4,3,2,4,'yes');
1 row created.

SQL> insert into applied values(5,3,2,1,'yes');
1 row created.

SQL>
```

```
SQL> select * from applied;
```

AID	SID	FID	CID	FAPPROVAL
1	1	1	1	yes
2	2	1	1	yes
3	2	1	2	yes
4	3	2	4	yes
5	3	2	1	yes

```
SQL>
```

Interactive Queries

```
select * from tabs;

select * from admin;

select * from student where sid > 3;

select * from COMPANIES natural join COMPANIES_BY_BRANCH;

select max(membership_id) from MEMBERSHIP_DETAILS ;

select avg(membership_id) from MEMBERSHIP_DETAILS ;
```

As the name suggests, the proposed project, the internship management system deals with managing all the details related to internships including application management, student and company data collection, and to real-time approval management so that the job of all the parties involved such as faculty, students and admin, becomes easier. This will radically help in reducing most of the pen and paperwork of students and faculty and will mostly be useful for management of internships at university level in the initial stages.

ooOoo