Name: Aarush Singh

Roll no: 201601107

Project: Hospital management system replica

Supervisor: H.K. Kamble (DGMIS)

Second Marker: M. Sukesh (AM)

Abstract

The main intention of introducing this system is to reduce the manual work at Health centre counters. Every sort of task is performed by the system, such as registering different types of persons (i.e patients, doctors and others) ,enquiries, and etc. reducing much paper work and burden of file storage. Also the latest information is right available for the officials and executives whenever and wherever they require. The system also facilitates us to know the types of tests a patient has taken in past. It also allows us to know about the patients treated under a doctor.

There are a lot of benefits to the Health centre by placing the system at their registration and enquiry counter. At the same time the patients are also benefited using this system. They can get the work done within no time.

Using the system is as simple as using the personal computer. Since end user computing is developing in our country, It is beneficial to both Health centre and the patients. Every step is clearly defined and help is provided through out the application of the formation of this project.

Acknowledgements

This project has taken a considerable amount of time and resources and I would like to acknowledge the help of all of those who have made the project possible. In particular I would like to thank my supervisor Sir H.K. Kamble for his time, patience and guidance, and also for allowing the idea to be pursued originally. I would also like to thank M. Sukesh Sir for his help, time and advice.

Further to these people I would like to thank the members of the Computer Support Group for their technical help in setting up various machines and telephone lines. Also, I would like to thank all of the many thousands of people who have worked on all of the Open Source projects without whose efforts this project would not have been possible.

Contents

About of the company

Indian Oil Corporation (IOC), India’s flagship national oil company and downstream petroleum major, was incorporated on June 30, 1959 as Indian Oil Company. The company was renamed as Indian Oil Corporation on September 1, 1964 following the merger of Indian Refineries (established 1958) with it.

The Indian Oil Group of companies owns and operates ten of India's 20 refineries with a combined refining capacity of 60.2 million metric tonnes per annum (MMTPA, .i.e. 1.2 million barrels per day). These include two refineries of subsidiary Chennai Petroleum Corporation. The corporation's cross–country network of crude oil and product pipelines, spanning over 10,000 km and the largest in the country, meets the vital energy needs of the consumers in an efficient, economical and environment–friendly manner.

IOC reaches precious petroleum products to millions of people everyday through a countrywide network of about 35,000 sales points. They are backed for supplies by 167 bulk storage terminals and depots, 101 aviation fuel stations and 89 Indane (LPG) bottling plants. About 7,335 bulk consumer pumps are also in operation for the convenience of large consumers, ensuring products and inventory at their doorstep.

The company operates the largest and the widest network of petrol and diesel stations in the country, numbering over 18,278. It reaches Indane cooking gas to the doorsteps of over 53 million households in nearly 2,700 markets through a network of about 5,000 Indane distributors.

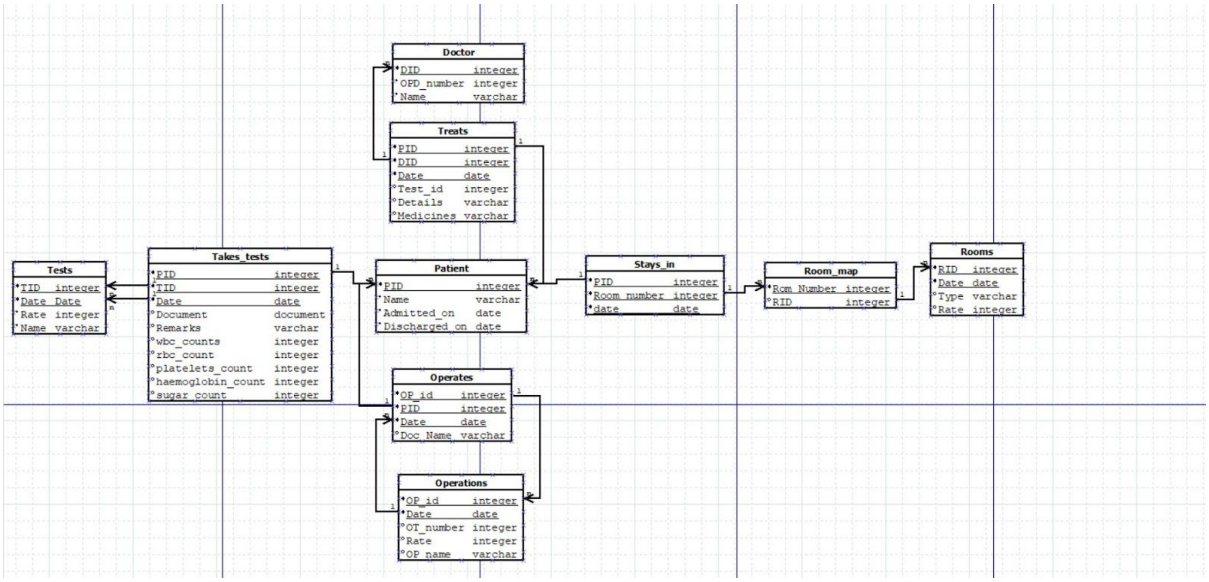
Indian Oil's ISO–9002 certified Aviation Service commands over 63% market share in the aviation fuel business, meeting the fuel needs of domestic and

international flag carriers, private airlines and the Indian Defence Services. The corporation also enjoys a dominant share of the bulk consumer business, including that of railways, state transport undertakings, and industrial, agricultural and marine sectors.

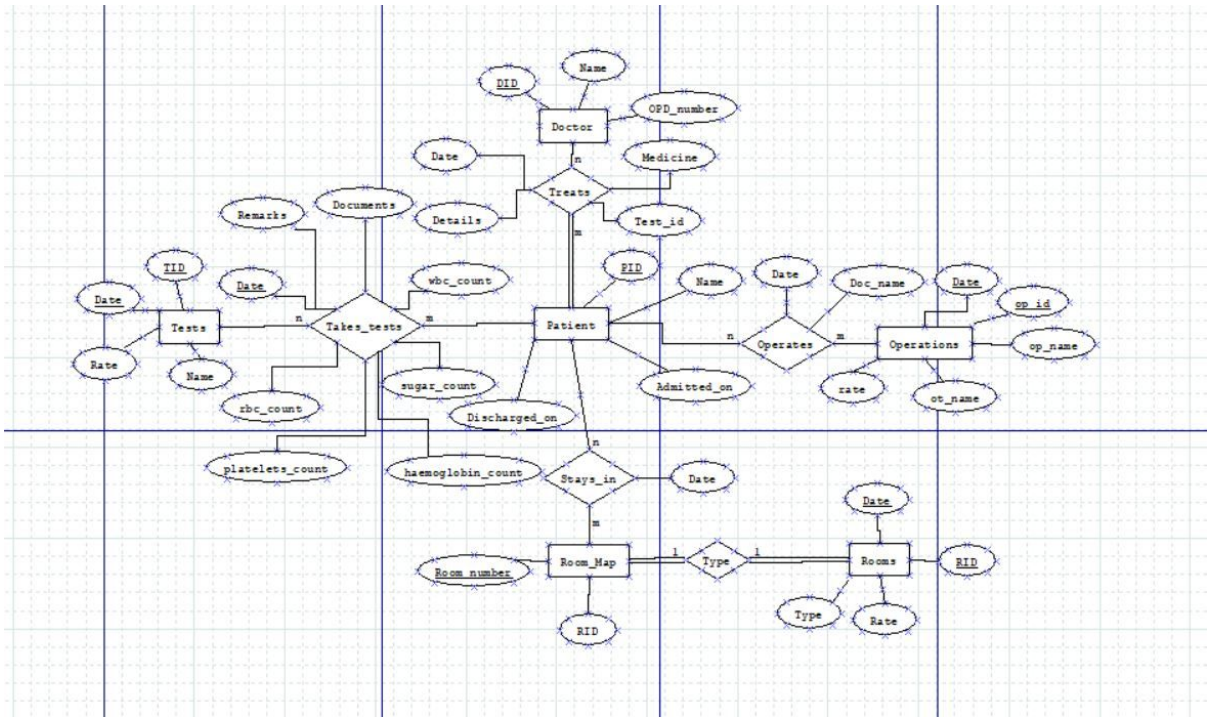
IOC's world–class R&D Centre is perhaps Asia's finest. Besides pioneering work in lubricants formulation, refinery processes, pipeline transportation and alternative fuels, the centre is also the nodal agency of the Indian hydrocarbon sector for ushering in hydrogen fuel economy in the country. It has set up a commercial Hydrogen–CNG station at an Indian Oil retail outlet in New Delhi this year. The centre holds 214 active patents, including 113 international patents.

IOC has joined the league of global technology providers with the selection of its in–house developed INDMAX technology (for maximising LPG yield) for the 4 MMTPA Fluidised Catalytic Cracking (FCC) unit at the Corporation's upcoming 15 MMTPA grass roots refinery at Paradip in Orissa, as well as for the FCC unit coming up at BRPL. A wholly–owned subsidiary, IndianOil Technologies, is engaged in commercialising the innovations and technologies developed by Indian Oil's R&D Centre.

ER Diagram


Doctor
*DID integer
*OPD_number integer
*Name varchar
Treats
*PID integer
*DID integer
*Date date
*Test_id integer
*Details varchar
*Medicines varchar
Tests
*TID integer
*Date_Date
*Rate integer
*Name varchar
Takes_tests
*PID integer
*TID integer
*Date date
*Document document
*Remarks varchar
*wbc_counts integer
*rbc_count integer
*platelets_count integer
*haemoglobin_count integer
*sugar_count integer
Patient
*PID integer
*Name varchar
*Admitted_on date
*Discharged_on date
Stays_in
*PID integer
*Room_number integer
*date date
Room_map
*Rom_Number integer
*RID integer
Rooms
*RID integer
*Date date
*Type varchar
*Rate integer
Operates
*OP_id integer
*PID integer
*Date date
*Doc_Name varchar
Operations
*OP_id integer
*Date date
*OT_number integer
*Rate integer
*OP_name varchar

Relational Schema



Functional dependencies and Normalization Proofs

TABLE: PATIENTS(pid,name,admitted_on,discharged_on)

pid-> name pid->admitted_on pid->discharged_on

KEY: pid
Normal Form: BCNF

TABLE: ROOMS(rid,_date,type,rate)

rid->type rid->rate

type->rid type->rate

KEY: rid / type Normal Form: BCNF

TABLE ROOM_MAP(room_number,rid)

room_number->rid

KEY: room_number Normal Form: BCNF

TABLE STAYS_IN(pid,room_number,_date)

{pid,date}->room_number

KEY: {pid,date} Normal Form: BCNF

TABLE: DOCTOR(did,name,opd_number)

did->name did->opd_number

KEY: did
Normal Form: BCNF

TABLE: TREATS(pid,did,medicines,test_id,details,_date)

{pid,did,_date}->medicines {pid,did,_date}->test_id {pid,did,_date}->details

KEY: {pid,did,_date} Normal Form: BCNF

TABLE: OPERATIONS(op_id,_date,op_name,ot_number,rate)

op_id->op_name {op_id,_date}->rate {op_id,_date}->ot_number

op_name->op_id {op_name,_date}->rate {op_name,_date}->ot_number

KEY: {op_id,_date} / {op_name,_date} Normal Form: BCNF

TABLE: OPERATES(op_id,pid,doc_name,_date)

{op_id,pid,_date}->doc_name

KEY: {op_id,pid,_date} Normal Form: BCNF

TABLE: TESTS(tid,_date,name,rate)

tid->name {tid,_date}->rate

name->tid {name,_date}->rate

KEY: {tid,_date} / {name,_date} Normal Form: BCNF

TABLE:

TAKES_TESTS(pid,tid,_date,document,remarks,wbc_count,rbc_count,haemoglobin_cou nt,sugar_count)

{pid,tid,_date}->document {pid,tid,_date}->remarks {pid,tid,_date}->wbc_count {pid,tid,_date}->rbc_count {pid,tid,_date}->haemoglobin_count {pid,tid,_date}->sugar_count

KEY: {pid,tid,_date} Normal Form: BCNF

DDL SCRPITS

PATIENT:

```
CREATE SCHEMA hospital;


CREATE TABLE Patient (
PID integer,
Name VARCHAR,
Admitted_on Date,
Discharged_on Date,
PRIMARY KEY (PID)
);
```

ROOMS:

```
CREATE TABLE Rooms (
RID integer,
_Date date,
Type varchar,
Rate integer,
PRIMARY KEY (RID,_Date)
);
```

ROOM_MAP:

```
create table Room_map(
Room_number integer,
RID integer,
PRIMARY KEY (Room_number)
);
```

STAYS_IN:

```
CREATE TABLE Stays_in (
PID integer,
Room_number integer,
_Date date,
PRIMARY KEY (PID,Room_number,_Date),
FOREIGN KEY (PID) REFERENCES Patient(PID)
ON DELETE SET DEFAULT ON UPDATE CASCADE,
FOREIGN KEY (Room_number) REFERENCES Room_map(Room_number)
ON DELETE SET DEFAULT ON UPDATE CASCADE
);
```

DOCTOR:

```
CREATE TABLE Doctor (
DID integer,
Name varchar,
OPD_number integer,
PRIMARY KEY (DID)
);
```

TREATS:

```
CREATE TABLE Treats (
PID integer,
DID integer,
Medicines varchar,
Test_ID integer,
Details varchar,
_Date date,
PRIMARY KEY (PID,DID,_Date),
FOREIGN KEY (PID) REFERENCES Patient(PID)
ON DELETE SET DEFAULT ON UPDATE CASCADE,
FOREIGN KEY (DID) REFERENCES Doctor(DID)
ON DELETE SET DEFAULT ON UPDATE CASCADE
);
```

OPERATES:

```
CREATE TABLE Operates (
OP_ID integer,
_Date date,
PID integer,
DID integer,
PRIMARY KEY (PID,OP_ID,_Date),
FOREIGN KEY (PID) REFERENCES Patient(PID)
ON DELETE SET DEFAULT ON UPDATE CASCADE,
FOREIGN KEY (OP_ID,_Date) REFERENCES Operations(OP_ID,_Date)
ON DELETE SET DEFAULT ON UPDATE CASCADE,
FOREIGN KEY (DID) REFERENCES Doctor(DID)
ON DELETE SET DEFAULT ON UPDATE CASCADE
);
```

OPERATIONS:

```
CREATE TABLE Operations (
OP_ID integer,
_Date date,
OP_Name varchar,
OT_Number integer,
Rate integer,
PRIMARY KEY (OP_ID,_Date)
);
```

TESTS:

```
CREATE TABLE Tests (
TID integer,
_Date date,
Name varchar,
Rate integer,
PRIMARY KEY (TID,_Date)
);
```

TAKES_TESTS:

```
CREATE TABLE Takes_Tests (
PID integer,
TID integer,
_Date date,
Document varchar,
Remarks varchar,
wbc_count integer,
rbc_count integer,
platelets_count integer,
haemoglobin_count float(20),
sugar_count integer,
PRIMARY KEY (PID,TID,_Date),
FOREIGN KEY (PID) REFERENCES Patient(PID)
ON DELETE SET DEFAULT ON UPDATE CASCADE,
FOREIGN KEY (TID,_Date) REFERENCES Tests(TID,_Date)
ON DELETE SET DEFAULT ON UPDATE CASCADE
);
```

STORED PROCEDURES

1.Given an op_id, give room number of patients currently staying in the hospital who have undergone this type of operation:

```
set search_path to hospital;
CREATE OR REPLACE FUNCTION func(OP_ID integer)
RETURNS TABLE(PID integer, Room_number integer) as $BODY$
DECLARE
        s integer;
        pd integer;
        rn integer;


BEGIN
        s:=op_id;

        SELECT op.pid into pd FROM OPERATES AS op JOIN (SELECT * FROM STAYS_IN natural JOIN PATIENT) AS si ON op.pid=si.pid
WHERE op.OP_ID=s and si.discharged_on is NULL;
        SELECT si.room_number into rn FROM OPERATES AS op JOIN (SELECT * FROM STAYS_IN natural JOIN PATIENT) AS si ON op.pid=si.pid
WHERE op.OP_ID=s and si.discharged_on is NULL;


        RETURN QUERY VALUES (pd,rn);
END
$BODY$ LANGUAGE plpgsql;
```

2.Lab Bill: Give total test charge of the patient named = ‘x’

```
set search_path to hospital;
CREATE OR REPLACE FUNCTION testcost(_name varchar)
RETURNS integer as $BODY$
DECLARE
        count integer;
BEGIN

        select sum(rate) into count from (select * from patient natural join takes_tests)  as a join tests on a.tid=tests.tid and a._date=tests._date where a.name=_name;


        RETURN count;
END
$BODY$ LANGUAGE plpgsql;
```

3.Total Cost of Patient:

```
set search_path to hospital;
CREATE OR REPLACE FUNCTION patientcost(_name varchar)
RETURNS integer as $BODY$
DECLARE
        count1 integer;
        count2 integer;
        count3 integer;
        total integer;
BEGIN
        total:=0;
        select sum(rate) into count1 from (select * from patient natural join takes_tests)  as a join tests on a.tid=tests.tid and a._date=tests._date where a.name=_name;
        select sum(rates) into count2 from (select * from operates natural join operations) as a natural join patient where name=_name group by name;
select sum(rate) into count3 from (select * from patient natural join (select * from stays_in
join (select room_number,_date as d,rate from rooms natural join room_map) as r on r.room_number=stays_in.room_number) as b
where b.d>=admitted_on and b.d<=discharged_on and patient.name=_name) as c;


        if count1 is not NULL then
                total=total+count1;
        end if;
        if count2 is not NULL then
                total=total+count2;
        end if;
        if count3 is not NULL then
                total=total+count3;
        end if;


        RETURN total;
END
$BODY$ LANGUAGE plpgsql
```

4.Total Income of hospital:

```
set search_path to hospital;
CREATE OR REPLACE FUNCTION totalincome(date1 date,date2 date)
RETURNS integer as $BODY$
DECLARE
        count1 integer;
        count2 integer;
        count3 integer;
        total integer;
        rec record;
        rec2 record;
        xyz date;
        abc integer;
BEGIN
        total:=0;
        for rec in select distinct _date from tests where _date>=date1 and _date<=date2
        loop
        xyz=rec._date;

        for rec2 in select distinct pid from patient
        loop
        abc=rec2.pid;
select sum(rate) into count1 from (select * from patient natural join (select * from stays_in
join (select room_number,_date as d,rate from rooms natural join room_map) as r on r.room_number=stays_in.room_number) as b
where b.d=xyz and b.d>=admitted_on and b.d<=discharged_on and patient.pid=abc) as c;
        select sum(rate) into count2 from (select * from patient natural join takes_tests)  as a join tests on a.tid=tests.tid and a._date=tests._date and a._date=xyz where a.pid=abc;
        select sum(rates) into count3 from (select * from operates natural join operations) as a natural join patient where pid=abc and a._date=xyz group by name;


        if count1 is not NULL then
                total=total+count1;
        end if;
        if count2 is not NULL then
                total=total+count2;
        end if;
        if count3 is not NULL then
                total=total+count3;
        end if;
        end loop;
        end loop;
        RETURN total;
END
$BODY$ LANGUAGE plpgsql
```

5.Given did tell the number of patients operated by the doctor:

```
set search_path to hospital;
CREATE OR REPLACE FUNCTION patient_count(did integer)
RETURNS integer as $BODY$
DECLARE
        t treats%rowtype;
        count integer;
BEGIN
count:=0;
FOR t IN SELECT * FROM TREATS
LOOP
IF t.did=did THEN
count:= count+1;
END IF;
END LOOP;
RETURN count;
END
$BODY$ LANGUAGE plpgsql;
```

STANDARD QUERIES

1. list the names of doctors who treated on ‘x’ patient.

SELECT doctor.name FROM(select * from patient natural join TREATS WHERE patient.Name='Paulie Banister') AS A JOIN DOCTOR on doctor.did=a.did;

2) list of patients and the ot_number in which they were operated on date d

SELECT distinct p.pid,p.name,b.ot_number FROM PATIENT AS p JOIN (SELECT ot_number,pid FROM OPERATIONS as op JOIN

(SELECT OP_ID,PID,_DATE FROM OPERATES WHERE _date='2017/11/12') AS A on op.op_id=a.op_id AND OP._DATE=A._DATE) AS B ON p.pid=B.pid;

3) list the rooms of the patient who had undergone an operation which had x operation id today

SELECT room_number FROM OPERATES AS op FULL JOIN (SELECT * FROM STAYS_IN natural JOIN PATIENT) AS si ON op.pid=si.pid

WHERE op.OP_ID=3 and discharged_on is NULL;

4) List the count of patients treated by a given doctor d

SELECT COUNT(PID) FROM TREATS where did=10 GROUP BY DID;

5) Give names of all patients treated by doctor “name” on date dt

SELECT a.NAME FROM (SELECT * FROM PATIENT JOIN TREATS ON PATIENT.PID=TREATS.PID) AS A JOIN

DOCTOR ON A.DID=DOCTOR.DID WHERE DOCTOR.name='Sisile Dow' and a._DATE='2017/11/05';

6) GIve the count of patients who were operated of op_id during a particular period

select count(pid) from (select pid from operates where _date>='2017/11/02' and _date<'2017/11/04' and op_id=1) as a;

7) Give the count and test name which is undergone by maximum patients

SELECT DISTINCT tests.name,abc.tid,abc.e FROM (Select tid,e from (SELECT tid,count(PID) as cnt FROM (SELECT * FROM TESTS natural JOIN TAKES_TESTS) as a GROUP BY TID) as d JOIN (select max(cnt) as e from (SELECT tid,count(PID) as cnt FROM (SELECT * FROM TESTS natural JOIN TAKES_TESTS) as a GROUP BY TID) as d) as z

ON d.cnt=z.e)as abc natural JOIN tests;

9) Give the total income of the hospital on a given date range

set search_path to hospital;

select sum(rate) from (select * from patient natural join (select * from stays_in

join (select room_number,_date as d,rate from rooms natural join room_map) as r on r.room_number=stays_in.room_number) as b

where b.d=xyz and b.d>=admitted_on and b.d<=discharged_on and patient.name='Paulie Banister') as c;

select sum(rate) from (select * from patient natural join takes_tests) as a join tests on a.tid=tests.tid and a._date=tests._date and a._date=xyz where a.name='Paulie Banister';

select sum(rates) from (select * from operates natural join operations) as a natural join patient where name='Paulie Banister' and a._date=xyz group by name;

10) Number of patients who were treated without being asked to do any tests

SELECT COUNT(PID) from (select pid FROM TREATS WHERE TEST_ID is NULL) as a;

11) Give the test reports of a given patient on a

select a.name, a._date, wbc_count, rbc_count, platelets_count, haemoglobin_count, sugar_count from (select * from patient natural join takes_tests) as a join tests on a.tid=tests.tid and a._date=tests._date where a.name='Paulie Banister' order by a._date desc;

12) Give the count of patients who obtained a dengue test as positive in a given time period:

select count(pid) from takes_tests where _date>='2017/11/05' and _date<='2017/11/06' and platelets_count<200000;

13) Give total stay charge of a patient named ‘x’:

set search_path to hospital;

select sum(rate) from (select * from patient natural join (select * from stays_in

join (select room_number,_date as d,rate from rooms natural join room_map) as r on r.room_number=stays_in.room_number) as b

where b.d>=admitted_on and b.d<=discharged_on and patient.name='Paulie Banister') as c;

14) Given a pid tell the type of room the patient is in:

select distinct type from (select * from rooms natural join room_map) as a join stays_in on stays_in.room_number=a.room_number where pid=1;

15) Total number of patients in any type of room:

select Type,count(PID) as no_of_patients from (select distinct pid,rid,type from (select * from (select * from rooms natural join room_map) as a join stays_in on stays_in.room_number=a.room_number) as b) as c group by type;

16) Names and Room numbers of patients of a particular doctor named “name” currently staying in rooms:

select name,room_number from (select * from stays_in natural join patient) as a join (select pid,name as xx from treats natural join doctor) as b on b.pid=a.pid where xx='Aidan Smeal' and discharged_on IS NULL;

17) Give total test charge of the patient named = ‘x’

select sum(rate) from (select * from patient natural join takes_tests) as a join tests on a.tid=tests.tid and a._date=tests._date where a.name='Paulie Banister';

18) Find the rate of operation of a patient named ‘x’:

select name,sum(rates) from (select * from operates natural join operations) as a natural join patient where name='Paulie Banister' group by name;

Drive link:
https://drive.google.com/file/d/1SRj6ho_xd5wiW-qD_j7p2OvNLjXZVy52/view?usp=sharing.