

**UJIAN TENGAH SEMESTER**

**DATA WAREHOUSE**

**GENAP 2020/2021**



Nama – Nama Anggota :

Maulana Taufik Hidayat (18082010053)

Dary Kharismojati (18082010057)

Deztra Irzhak Pratama (18082010071)

**Program Studi Sistem Informasi**

**Fakultas Ilmu Komputer**

**Universitas Pembangunan Nasional “Veteran” Jawa Timur**

**Tahun 2021**

Berdasarkan database operasional rumah sakit di atas!

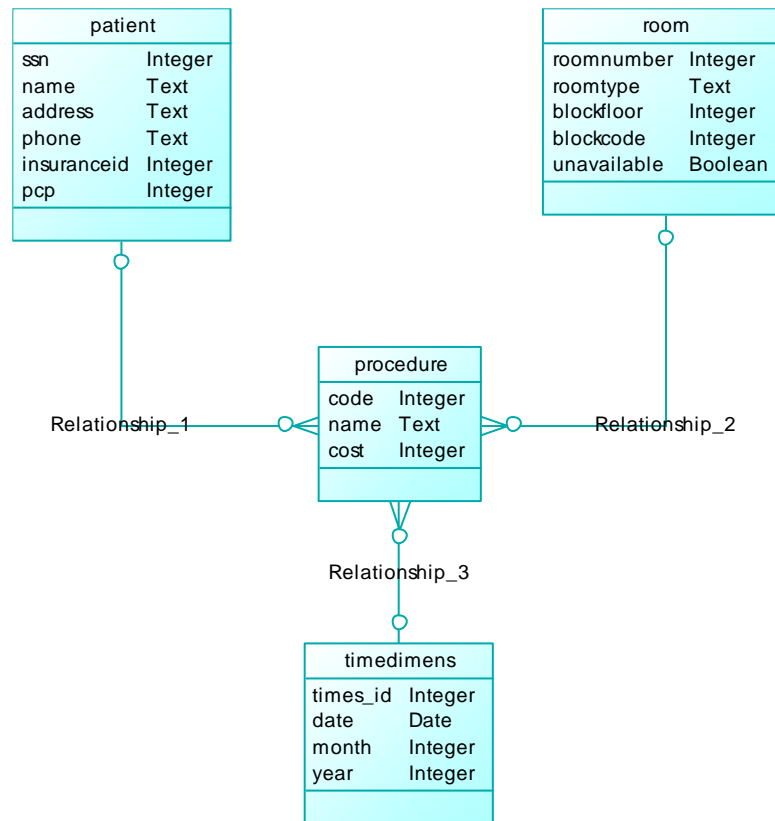
1. Identifikasi kandidat *measure* dari database di atas, kemudian buatlah 4 (empat) buah pertanyaan yang bersifat analitik (multidimensi) dari manajer (*decision maker*) rumah sakit!
2. Untuk menjawab semua pertanyaan tersebut kemudian:
  - a. Menggunakan langkah-langkah pemodelan multidimensi, rancanglah model multidimensi untuk Data Warehouse dari database operasional di atas!
  - b. Buatlah kalimat untuk memeriksa kevalidan model multidimensi anda!
3. Implementasikan rancangan model multidimensi menggunakan MySql lengkap dengan atribut-atributnya.
4. Buat rancangan tampilan OLAP dari Data Warehouse anda untuk menjawab 4 (empat) pertanyaan manajer pada nomor 1 !
5. Tuliskan query SQL untuk menjawab 4 (empat) pertanyaan pada nomor 1 menggunakan
  - a) Database operasional (gambar 1 di atas)
  - b) Datawarehouse yang anda rancang
6. Tuliskan jawaban anda termasuk gambar (capture) model multidimensi anda di MS Word, berikan Cover Standar (berjudul UTS DWO Genap 2020/2021, nama-nama anggota, prodi, tahun, dll). Simpan dengan nama “UTSGenap2020-ParA/B-NPMKetua.docx”, konversi ke PDF dengan nama yang sama, submit oleh ketua kelompoknya melalui link yang disediakan di elearning.

Jawaban :

1. Identifikasi kandidat pada tabel database adalah cost,  
Pertanyaan :
  - Berapa banyak jumlah pasien yang menetap di ruangan ?
  - Ruangan tipe apa yang paling banyak di isi oleh pasien ?
  - Berapa pasien yang paling banyak mengisi ruangan pada bulan mei 2021 ?
  - Blockfloor berapa yang paling banyak digunakan pasien pada bulan April 2021?

2.

A.



B.

- Jumlah pasien yang menetap diruangan dapat dilihat dari SSN yang dihubungkan pada procedure yang terhubung dengan room
- Tipe ruangan terbanyak yang diisi oleh pasien dapat dilihat dari roomtype yang terhubung pada procedure dan SSN pasien

3.

create table PATIENT

```
(
  SSN          int not null,
  NAME         text,
  ADDRESS      text,
```

```
        PHONE          text,  
        INSURANCEID    int,  
        PCP            int,  
        primary key (SSN)  
    );
```

create table "PROCEDURE"

```
(  
    SSN            int,  
    TIMES_ID       int,  
    ROOMNUMBER     int,  
    CODE           int,  
    NAME           text,  
    COST           int  
);
```

create table ROOM

```
(  
    ROOMNUMBER     int not null,  
    ROOMTYPE       text,  
    BLOCKFLOOR     int,  
    BLOCKCODE      int,  
    UNAVAILABLE    bool,  
    primary key (ROOMNUMBER)  
);
```

create table TIMEDIMENS

```
(  
    TIMES_ID       int not null,  
    DATE           date,  
    MONTH          int,  
    YEAR           int,
```

primary key (TIMES\_ID)

);

alter table "PROCEDURE" add constraint FK\_RELATIONSHIP\_1 foreign key (SSN)

references PATIENT (SSN) on delete restrict on update restrict;

alter table "PROCEDURE" add constraint FK\_RELATIONSHIP\_2 foreign key (ROOMNUMBER)

references ROOM (ROOMNUMBER) on delete restrict on update restrict;

alter table "PROCEDURE" add constraint FK\_RELATIONSHIP\_3 foreign key (TIMES\_ID)

references TIMEDIMENS (TIMES\_ID) on delete restrict on update restrict;

4.

TIME	INSURANCE	ROOM	
		A	B
03 05 21	Insurance Cost	300	200
	Regular Cost	200	100
14 02 21	Insurance Cost	300	200
	Regular Cost	100	150
12 01 21	Insurance Cost	500	400
	Regular Cost	300	200
06 03 21	Insurance Cost	250	150
	Regular Cost	150	100
09 10 21	Insurance Cost	500	200
	Regular Cost	200	100

5. A. - select count(pa.\*) from procedure p join undergoes u (p.code=u.procedure) join stay s (u.stay=s.stayid) join room r (s.room=r.roomnumber) join patient pa (u.patient=patient.ssn)

- select max(count(p.\*, pa.\*, r.\*, t.\*)) from procedure p join undergoes u (p.code=u.procedure) join stay s (u.stay=s.stayid) join room r (s.room=r.roomnumber) join patient pa (u.patient=patient.ssn) group by r.roomtype
- select count(pa.\*) from procedure p join undergoes u (p.code=u.procedure) join stay s (u.stay=s.stayid) join room r (s.room=r.roomnumber) join patient pa (u.patient=patient.ssn) where s.start\_date between '1-may-2021' and '31-may-2021'
- select max(count(p.\*, pa.\*, r.\*, t.\*)) procedure p join undergoes u (p.code=u.procedure) join stay s (u.stay=s.stayid) join room r (s.room=r.roomnumber) join patient pa (u.patient=patient.ssn) where s.start\_date between '1-apr-2021' and '30-apr-2021' group by r.blockfloor

B.

- select count(pa.\*) from procedure p join patient pa(p.ssn=pa.ssn) join room r (p.roomnumber=r.roomnumber).
- select max(count(p.\*, pa.\*, r.\*, t.\*)) from procedure p join patient pa(p.ssn=pa.ssn) join room r (p.roomnumber=r.roomnumber) group by r.roomtype
- select count(pa.\*) from procedure p join patient pa(p.ssn=pa.ssn) join room r (p.roomnumber=r.roomnumber) join timedimens t (p.times\_id=t.times\_id) where t.month=5 and t.year=2021
- select max(count(p.\*, pa.\*, r.\*, t.\*)) from procedure p join patient pa(p.ssn=pa.ssn) join room r (p.roomnumber=r.roomnumber) join timedimens t (p.times\_id=t.times\_id) where t.month=4 and t.year=2021 group by r.blockfloor