

Reporte DB

Entregas programadas:

Viernes 23/2 :

Análisis de dominio
Sistema de entidad/relación

Viernes 2/3:

ER
Stack de tecnología
Entorno montado y VCS (sistema de computación visual)

Viernes 9/3:

Prototipo módulo 1

Viernes 16/3

Prototipo módulo 2

Viernes 23/3

Entrega

Primera entrega

Temario A

23/02/2018

Módulo 1

Análisis de dominio

Cambio:

- **La receta asociada a la visita**
- **1 recetas muchos medicamentos**
- **Medicamento → entidad**
- ¿Qué entidades?
 - Médico y paciente
- ¿Qué relaciones?
 - Paciente visita médico
 - Médico examina paciente
 - Médico realiza receta
- **Análisis de requerimientos:** que necesitamos guardar, que vamos a analizar, quien accede a los datos, cuáles serán las consultas más frecuentes, entre otras.
 - Guardar: nombre y apellidos del paciente, antecedentes médicos, edad, sexo, religión, enfermedad, medicamento recetado, teléfono, domicilio, sarampión, fecha de nacimiento, peso, talla, número de

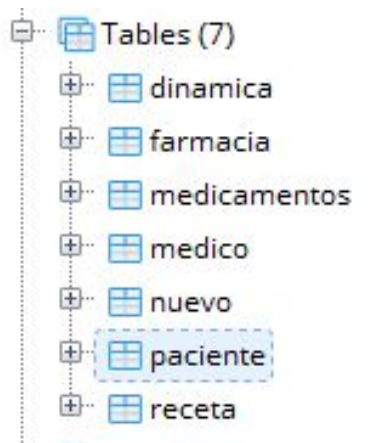
visita, fecha de consulta, nombre del médico, especialidad, id de las entidades.

- Analizar:
 - Cantidad de pacientes por día, semana, mes y año
 - Identificar qué día es el más concurrido
 - Por especialidad del médico
 - Enfermedades frecuentes
 - Con base a edad
 - Con base a sexo
 - Medicamentos frecuentes
 - De que departamento hay más pacientes
- Accede a los datos únicamente el médico que realizó la consulta.
- Consultas más frecuentes:
 - Historial de los pacientes
 - Cantidad de pacientes
- Diseño conceptual: Descripción de la BD en alto nivel; Su objetivo es ser lo suficientemente técnico para poder trabajar a partir de él y lo suficientemente abstracto para que el cliente los comprenda.
- Aspectos lógicos, físicos y de seguridad.

Tablas extras:

- Tabla de recetas
- Tabla de medicamentos
- Tabla de farmacias
- Tabla de contacto de médicos externos

Descripción de las tablas



- **Tabla medico**
 - Listado de los médicos y sus datos principales
 - medico(id_medico serial PRIMARY KEY,

nombre VARCHAR,
especialidad VARCHAR);

- **Tabla paciente**

- En esta tabla se guardan todos los pacientes con sus respectivos atributos.

PostgreSQL 10 - Proyecto1P - public.paciente

```

1 SELECT * FROM public.paciente
2 ORDER BY id_paciente ASC

```

	nombre character (25)	apellidos character (30)	Peso double precision	id_paciente [PK] integer	Sexo boolean	Teléfono character (12)	Departamento character (100)	Domicilio character (100)	Religión character (50)
1	Samantha	Duarte	[null]	21	[null]	[null]	[null]	[null]	[null]
2	Ivette	Cardona	[null]	22	[null]	[null]	[null]	[null]	[null]
3	Samantha	Duarte	[null]	23	[null]	[null]	[null]	[null]	[null]
4	Ivette	Cardona	[null]	24	[null]	[null]	[null]	[null]	[null]
5	Samantha	Duarte	[null]	25	[null]	[null]	[null]	[null]	[null]
6	Ivette	Cardona	[null]	26	[null]	[null]	[null]	[null]	[null]
7	Samantha	Duarte	[null]	27	[null]	[null]	[null]	[null]	[null]
8	Ivette	Cardona	[null]	28	[null]	[null]	[null]	[null]	[null]
9	Samantha	Duarte	[null]	29	[null]	[null]	[null]	[null]	[null]
10	Ivette	Cardona	[null]	30	[null]	[null]	[null]	[null]	[null]

Entre los atributos se encuentran:

paciente

General Columns Constraints Advanced Parameter Security SQL

Inherited from table(s)

	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	nombre	character	25		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	apellidos	character	30		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	Peso	double precision			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_paciente	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	Sexo	boolean			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	Teléfono	character	12		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	Departamento	character	100		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	Domicilio	character	100		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	Religión	character	50		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	username	character varying			<input type="checkbox"/> No	<input type="checkbox"/> No

- **Tabla receta**

- Guarda las recetas y su contenido

- **Tabla medicamentos**

- Contiene la información de todos los medicamentos correspondientes a las casas médicas disponibles.

- medicamentos(id_medimento serial PRIMARY KEY,
nombre VARCHAR,
casa_medica VARCHAR
fecha_venc DATE,
precio FLOAT);

- **Tabla farmacia**

- Contiene información de las farmacias que proveen los medicamentos que los médicos recetan

- **Tabla nuevo**

- Guarda la información necesaria para crear un nuevo atributo en una tabla.
- nuevo(id_tabla serial PRIMARY KEY,

atributo VARCHAR,
nombre_tabla VARCHAR);

- **Tabla dinamica**

- Se encarga de actualizar los atributos añadidos en la tabla nuevo
 - dinamica(id_tabla serial PRIMARY KEY,
atributo VARCHAR);

Queries probados

- **Tabla Médicos**

```
SELECT id_medico as id, nombre, especialidad FROM "Medico"  
INSERT INTO "Medico" (nombre, especialidad) VALUES ($1, $2)  
UPDATE "Medico" SET nombre = $1, especialidad = $2 WHERE id_medico = $3  
DELETE FROM "Medico" WHERE id_medico = $1  
SELECT id_medico as id, nombre, especialidad FROM "Medico" WHERE $1 = $2
```

- **Tabla Farmacia**

```
SELECT * FROM farmacia  
INSERT INTO farmacia (nombre, telefono) VALUES ($1, $2)  
UPDATE farmacia SET nombre = $1, telefono = $2 WHERE id_farmacia = $3  
DELETE FROM farmacia WHERE id_farmacia = $1  
SELECT * FROM farmacia WHERE $1 = $2
```

- **Tabla Medicamentos**

```
SELECT * FROM medicamentos  
INSERT INTO medicamentos (nombre) VALUES ($1)  
UPDATE medicamentos SET nombre = $1 WHERE id_medicamentos = $3  
DELETE FROM medicamentos WHERE id_medicamentos = $1  
SELECT * FROM farmacia WHERE $1 = $2
```

Extras

¿Cómo añadir atributos a una tabla ya existente?

```
1. ALTER TABLE member  
2. ADD e_mail CHAR(20);  
3.  
4. UPDATE member SET e_mail='johnwhite@msn.com' WHERE member_no='A2345';  
5. UPDATE member SET e_mail='leeJulie@aol.com' WHERE member_no='R3456';  
6. UPDATE member SET e_mail='d_ford@yahoo.com' WHERE member_no='S4567';  
7. UPDATE member SET e_mail='maryhowe@aol.co.uk' WHERE member_no='S7654';  
8. UPDATE member SET e_mail='mrobinson@msn.com' WHERE member_no='A7412';  
9. UPDATE member SET e_mail='ahmed_h@msn.co.uk' WHERE member_no='B6421';  
10. UPDATE member SET e_mail='AnnMat@f2s.com' WHERE member_no='A1246';  
11. UPDATE member SET e_mail='amohammady@box.com' WHERE member_no='R8916';  
12. UPDATE member SET e_mail='fairyjack@aol.com' WHERE member_no='S1248';
```

First add Age column in table1

```
ALTER TABLE table1 ADD COLUMN Age TINYINT UNSIGNED DEFAULT 0;
```

then update that column using blow query

```
UPDATE table1 t1  
INNER JOIN Tabel2 t2 ON t1.id = t2.id  
SET t1.age = t2.age;
```

Fuentes de referencia

- https://www.ibm.com/support/knowledgecenter/es/SSZJPZ_9.1.0/com.ibm.swg.im.iis.ia.application.doc/topics/r_dq_domainanalysis.html
- <https://stackoverflow.com/questions/27376152/how-to-add-a-column-to-a-table-from-another-table-in-mysql>