

Diplomatura en Business Analytics

Evaluación Final Módulo 3 - ETL y Datawarehouse

Fabián García y Mariana Grandis



Desarrollo

1. Agregar al modelo pre-existente las tablas que sean necesarias para el correcto funcionamiento de un modelo de streaming on demand (distribución digital de contenido multimedia a través de una red de computadoras a demanda del usuario, por ejemplo: Spotify, YouTube, HBO Max, Netflix). Al menos deberán ser insertadas 3 tablas.
2. Insertar las tablas en el diagrama indicando claves, relaciones y cualquier otro detalle relevante
3. Eliminar, de ser necesario, las tablas sobrantes. Justificar debidamente esa decisión (tanto sea si se deciden borrar tablas como si no)
4. Crear dentro de la base de datos en PostgreSQL las tablas detalladas en el punto 1
5. Ingresar datos ficticios pero verosímiles en las tablas creadas (por ejemplo, no cargar un campo nombre con "AAAA" sino con "Federico", o un campo numérico con 12345678 sino con 4054)
6. Generar al menos dos vistas/consultas que permitan realizar un análisis del negocio (mencionar que análisis se quiere realizar con la vista/consulta). La vista debe ser en

formato tabla, ser clara y concisa y debe poder ser interpretada de forma inequívoca. Por ejemplo, una tabla repleta de códigos / ID no permite identificar nada. Una tabla con nombres, títulos, precios, valores sumariados por categoría, sí puede ser interpretada.

Material a entregar

- Nueva estructura de la base de datos (diagrama DER anterior más las modificaciones realizadas durante el TP)
- Todos los comandos SQL ejecutados (consultas, inserciones, creaciones de tablas) en archivo .sql o Word
- Un documento donde se detallen y justifiquen las modificaciones realizadas. Deberá contener también una captura de pantalla de las vistas generadas
- Todas las consultas/vistas generadas deberán ser justificadas en cuanto a su motivo de creación y aprovechamiento futuro en la empresa

1) Nuevas tablas:

Elegimos para el funcionamiento de un modelo de streaming, la plataforma Netflix, en donde agregaríamos las siguientes 4 tablas:

Plan Type: la creamos para diferenciar los tipos de planes disponibles del nuevo servicio.

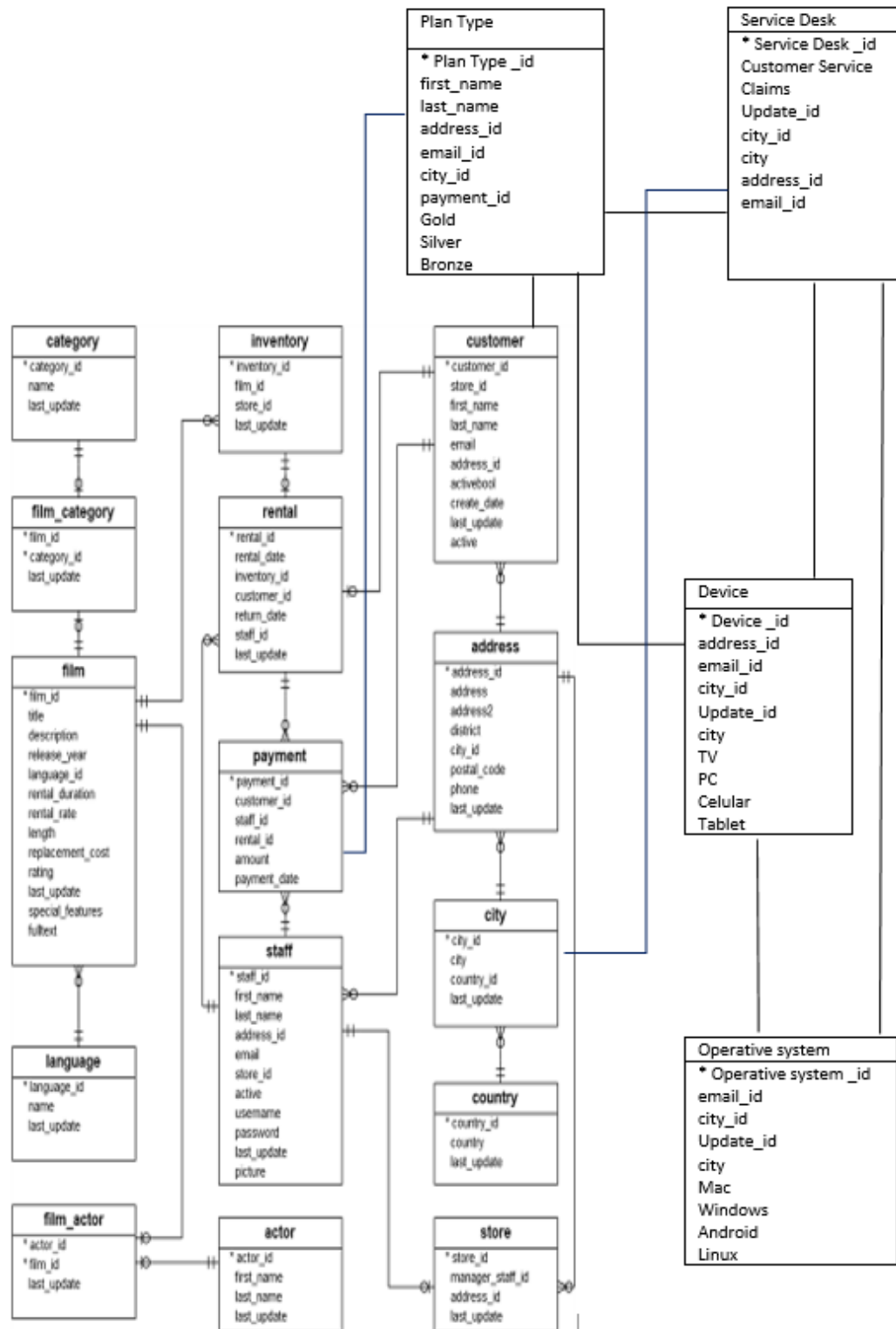
Device: la creamos para identificar los tipos posible de dispositivos de los usuarios.

Operative system: al crear Device, vimos la necesidad de vincularlo a un sistema operativo (seguridad).

Service: como medio de asistencia sincrónica o asincrónica al cliente.

Plan Type <u>* Plan Type_id</u> first_name last_name address_id email_id city_id payment_id Gold Silver Bronze	Service Desk <u>* Service Desk_id</u> Customer Service Claims Update_id city_id city address_id email_id	Device <u>* Device_id</u> address_id email_id city_id Update_id city TV PC Celular Tablet	Operative system <u>* Operative system_id</u> email_id city_id Update_id city Mac Windows Android
---	---	--	--

2. Insertar las tablas en el diagrama indicando claves, relaciones y cualquier otro detalle relevante



3. Eliminar, de ser necesario, las tablas sobrantes. Justificar debidamente esa decisión (tanto sea si se deciden borrar tablas como si no)

Decidimos que no eliminamos ninguna tabla. Todas son necesarias para el nuevo modelo o se perdería trazabilidad o historial necesaria para mantener la calidad del nuevo servicio. La tabla Store, la mantenemos porque proponemos que las tiendas de Blockbooster, se transformen en las nuevas bases de trabajo para la nueva estructura.

4. Crear dentro de la base de datos en PostgreSQL las tablas detalladas en el punto 1

Plan Type
* Plan Type_id
first_name
last_name
address_id
email_id
city_id
payment_id
Gold
Silver
Bronze

```
CREATE TABLE Plan_type
(Plan_type int primary key,
first_name varchar(50)not null, last_name varchar(20) not null,
address_id varchar(50) not null,email varchar(20) not null,
city_id varchar(50) not null,payment_id varchar(20) not null,
Gold varchar(20) not null, Silver varchar (20)not null,
Bonze varchar(20) not null)
```

```
CREATE TABLE Plan_type
(Plan_type int primary key,
first_name varchar(50)not null, last_name varchar(20) not null,
address_id varchar(50) not null,email varchar(20) not null,
city_id varchar(50) not null,payment_id varchar(20) not null,
Gold varchar(20) not null, Silver varchar (20)not null,
Bonze varchar(20) not null)
```

▼	plan_type
▼	Columns (10)
	plan_type
	first_name
	last_name
	address_id
	email
	city_id
	payment_id
	gold
	silver
	bronze

Device
* Device_id
address_id
email_id
city_id
Update_id
city
TV
PC
Celular
Tablet

CREATE TABLE Device

(Device_id int primary key,

address_id varchar(50) not null, email_id varchar(50) not null,

city_id varchar(50) not null, city varchar(50) not null,

Update_id varchar(50) not null,

TV varchar(20) not null, PC varchar (20)not null,

Celular varchar(20) not null,

Tablet varchar (20))

```
CREATE TABLE Device
(Device_id int primary key,
address_id varchar(50) not null, email_id varchar(50) not null,
city_id varchar(50) not null, city varchar(50) not null,
Update_id varchar(50) not null,
TV varchar(20) not null, PC varchar (20)not null,
Celular varchar(20) not null,
Tablet varchar (20))
```

device
Columns (10)
device_id
address_id
email_id
city_id
city
update_id
tv
pc
celular
tablet

Operative system
* Operative system _id
email_id
city_id
Update_id
city
Mac
Windows
Android

CREATE TABLE Operative_System

(Operative_system_id int primary key,

email_id varchar(20) not null,city_id varchar(20) not null,

Update_id varchar(20) not null, city varchar(50) not null,

Mac varchar(20) not null, Windows varchar (20)not null,

Android varchar(20) not null,

Linux varchar (20))

```
CREATE TABLE Operative_System
(Operative_system_id int primary key,
email_id varchar(20) not null,city_id varchar(20) not null,
Update_id varchar(20) not null, city varchar(50) not null,
Mac varchar(20) not null, Windows varchar (20)not null,
Android varchar(20) not null,
Linux varchar (20))
```

operative_system
Columns (9)
operative_system_id
email_id
city_id
update_id
city
mac
windows
android
linux

Service Desk
* Service Desk_id
Customer Service
Claims
Update_id
city_id
city
address_id
email_id

CREATE TABLE Service_Desk

(Service_Desk_id int primary key,

customer_service varchar(50)not null,

claims numeric not null,Update_id varchar(50) not null,city_id varchar(50) not null,

city varchar(50) not null,address_id varchar(50) not null,

Up_date numeric not null,email_id varchar(50) not null)

CREATE TABLE Service_Desk

(Service_Desk_id int primary key,

customer_service varchar(50)not null,

claims numeric not null,Update_id varchar(50) not null,city_id varchar(50) not null,

city varchar(50) not null,address_id varchar(50) not null,

Up_date numeric not null,email_id varchar(50) not null)

service_desk
Columns (9)
service_desk_id
customer_service
claims
update_id
city_id
city
address_id
up_date
email_id

5. Ingresar datos ficticios pero verosímiles en las tablas creadas (por ejemplo, no cargar un campo nombre con “AAAA” sino con “Federico”, o un campo numérico con 12345678 sino con 4054)

Plan_Type
* Plan_Type_id
first_name
last_name
address_id
email_id
city_id
payment_id
Gold
Silver
Bronze

INSERT INTO Plan_Type

VALUES

(1981,'pablo','perez','los escritores34','pabloperez','lujan','visa','si','no','no')

INSERT INTO Plan_type

VALUES

(1980,'juan','lopez','los alamos134','juanlopez','junin','mastecard','no','si','no')

INSERT INTO Plan_Type

VALUES

(1983,'mariana','lopez','los reyunos121','marianalopez','ciudad','visa','no','no','si')

SELECT *

FROM Plan_Type

```

34 INSERT INTO Plan_Type
35 VALUES
36 (1981,'pablo','perez','los escritores34','pabloperez','lujan','visa','si','no','no')
37
38 INSERT INTO Plan_type
39 VALUES
40 (1980,'juan','lopez','los alamos134','juanlopez','junin','mastecard','no','si','no')
41
42 INSERT INTO Plan_Type
43 VALUES
44 (1983,'mariana','lopez','los reyunos121','marianalopez','ciudad','visa','no','no','si')
45
46 SELECT *
47 FROM Plan_Type
48
49
50 INSERT INTO Device
51 VALUES
52 (300,'los escritores34','pabloperez','lujan','si','pedriel','si','no','no','si')
53

```

Data Output Explain Messages Notifications

	plan_type [PK] integer	first_name character varying (50)	last_name character varying (20)	address_id character varying (50)	email character varying (20)	city_id character varying (50)	payment_id character varying (20)	gold character varying (20)	silver character varying (20)	bronze character
1	1981	pablo	perez	los escritores34	pabloperez	lujan	visa	si	no	no
2	1977	juan	lopez	los alamos134	juanlopez	junin	mastecard	no	si	no

Device
* Device_id
address_id
email_id
city_id
Update_id
city
TV
PC
Celular
Tablet

INSERT INTO Device

VALUES

(300,'los escritores34','pabloperetz','lujan','si','pedriel','si','no','no','si')

INSERT INTO Device

VALUES

(301,'los reyunos121','marianalopez','ciudad','no','mendoza','si','si','si','si')

INSERT INTO Device

VALUES

(302, 'los alamos134','juanlopez','junin','no','mendoza','si','si','si','si')

SELECT*

FROM Device

```

50 INSERT INTO Device
51 VALUES
52 (300,'los escritores34','pabloperetz','lujan','si','pedriel','si','no','no','si')
53
54 INSERT INTO Device
55 VALUES
56 (301,'los reyunos121','marianalopez','ciudad','no','mendoza','si','si','si','si')
57
58 INSERT INTO Device
59 VALUES
60 (302, 'los alamos134','juanlopez','junin','no','mendoza','si','si','si','si')
61
62 SELECT*
63 FROM Device

```

Data Output Explain Messages Notifications

	device_id [PK] integer	address_id character varying (50)	email_id character varying (50)	city_id character varying (50)	city character varying (50)	update_id character varying (50)	tv character varying (20)	pc character varying (20)	celular character varying (20)	tablet character v
1	300	los escritores34	pabloperetz	lujan	si	pedriel	si	no	no	si
2	301	los reyunos121	marianalopez	ciudad	no	mendoza	si	si	si	si
3	302	los alamos134	juanlopez	junin	no	mendoza	si	si	si	si

Operative system
* Operative system_id
email_id
city_id
Update_id
city
Mac
Windows
Android










```
INSERT INTO Operative_System
VALUES
(304,'pabloperez','lujan','si','no','si','si','no','no')

INSERT INTO Operative_System
VALUES
(305, 'juanlopez','junin','si','si','si','no','si','no')

INSERT INTO Operative_System
VALUES
(302, 'marianalopez','mendoza','si','si','si','si','si','si')

SELECT*
FROM Operative_System
```

```
65 INSERT INTO Operative_System
66 VALUES
67 (304,'pabloperez','lujan','si','no','si','si','no','no')
68
69 INSERT INTO Operative_System
70 VALUES
71 (305, 'juanlopez','junin','si','si','si','no','si','no')
72
73 INSERT INTO Operative_System
74 VALUES
75 (302, 'marianalopez','mendoza','si','si','si','si','si','si')
76
77 SELECT*
78 FROM Operative_System
79
```

Data Output		Explain	Messages	Notifications					
	 operative_system_id [PK] integer	 email_id character varying (20)	 city_id character varying (20)	 update_id character varying (20)	 city character varying (50)	 mac character varying (20)	 windows character varying (20)	 android character varying (20)	 linux character varying (20)
1	300	pabloperez	lujan	si	no	si	si	no	[null]
2	301	marianalopez	mendoza	si	si	si	si	si	[null]
3	303	juanlopez	junin	si	si	si	no	si	[null]
4	302	marianalopez	mendoza	si	si	si	si	si	si

Service Desk
* Service Desk_id
Customer Service
Claims
Update_id
city_id
city
address_id
email_id

INSERT INTO Service_Desk

VALUES

(300,'si',10,'si','lujan','pedriel','los escritores34',12,'pabloperrez')

INSERT INTO Service_Desk

VALUES

(310,'no',112,'si','junin','junin','los alamos134',14,'juanlopez')

INSERT INTO Service_Desk

VALUES

(320,'si',120,'si','ciudad','mendoza','los reyunos121',14,'marianalopez')

SELECT*

FROM Service_Desk

```

81 INSERT INTO Service_Desk
82 VALUES
83 (300,'si',10,'si','lujan','pedriel','los escritores34',12,'pabloperrez')
84
85 INSERT INTO Service_Desk
86 VALUES
87 (310,'no',112,'si','junin','junin','los alamos134',14,'juanlopez')
88
89 INSERT INTO Service_Desk
90 VALUES
91 (320,'si',120,'si','ciudad','mendoza','los reyunos121',14,'marianalopez')
92
93 SELECT*
94 FROM Service_Desk
95

```

	service_desk_id [PK] integer	customer_service character varying (50)	claims numeric	update_id character varying (50)	city_id character varying (50)	city character varying (50)	address_id character varying (50)	up_date numeric	email_id character varying (50)
1	300	si	10	si	lujan	pedriel	los escritores34	12	pabloperrez
2	310	no	112	si	junin	junin	los alamos134	14	juanlopez
3	320	si	120	si	ciudad	mendoza	los reyunos121	14	marianalopez

- Generar al menos dos vistas/consultas que permitan realizar un análisis del negocio (mencionar que análisis se quiere realizar con la vista/consulta). La vista debe ser en formato tabla, ser clara y concisa y debe poder ser interpretada de forma inequívoca.

La tabla a continuación hace referencia a la cantidad de veces que reclaman los clientes

SELECT t3.claims as reclamo,t2.last_name as apellido,t2.first_name as nombre, t2.email
FROM plan_type as t2
JOIN service_desk as t3
on t3.address_id = t2.address_id
JOIN city as t1
on t2.city_id = t3.city_id

```
106 SELECT t3.claims as reclamo,t2.last_name as apellido,t2.first_name as nombre, t2.email
107 FROM plan_type as t2
108 JOIN service_desk as t3
109 on t3.address_id = t2.address_id
110 JOIN city as t1
111 on t2.city_id = t3.city_id
112
```

	reclamo	apellido	nombre	email
	numeric	character varying (20)	character varying (50)	character varying (20)
1	10	perez	pablo	pabloperez

Esta tabla toma como manera univoca el número de reclamo de los clientes.

```
103 SELECT t1.claims,t2.first_name as nombre, t2.last_name as apellido
104 FROM service_desk as t1
105 JOIN plan_type as t2
106 on t1.city_id = t2.city_id
107 JOIN device as t3
108 on t3.email_id = t1.email_id
109
```

	claims	nombre	apellido
	numeric	character varying (50)	character varying (20)
1	10	pablo	perez
2	112	juan	lopez
3	112	juan	lopez
4	120	mariana	lopez

