**Proyecto – Iteración 4: Alohandes**

Angela M. Vargas Guerrero, Gabriela Caballero Duran

ISIS 2304 – Sistemas Transaccionales

Universidad de los Andes, Bogotá, Colombia

{am.vargasg, g.caballero[}@uniandes.edu.co](mailto:%7d@uniandes.edu.co)

Fecha de presentación: Mayo 26 de 2020

**Tabla de contenido**

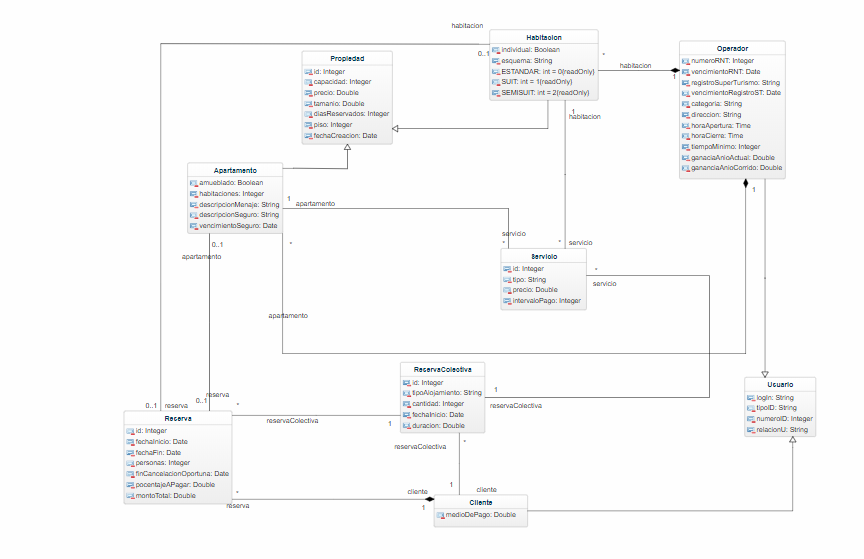
1. **Introducción**

En este documento se presentan los modelos conceptuales y transaccionales propuestos a partir de la descripción del modelo del negocio AlohAndes. Adicionalmente se presenta un resumen con los requerimientos funcionales más importantes en los cuales la integridad de datos es fundamental y finalmente los diagramas de secuencia que muestran cómo se satisfacen estos requerimientos a partir del modelo planteado.

AlohAndes pretende ser una aplicación informática que sirva de intermediario para conseguir y ofertar diversas opciones de alojamiento.

1. **Modelo Conceptual**

A continuación, se presenta el modelo conceptual (UML) donde se identifican las entidades relevantes, sus características y las relaciones que entre ellas tienen lugar.



Del modelo anterior las clases operador, apartamento, habitación y reserva necesitan ser persistentes pues la información almacenada en estas se introduce una vez y debe estar disponible cada vez que se abre la aplicación.

1. **Modelo de Datos Relacional**
   1. **Ahora se muestra el modelo de datos relacional con base en el modelo conceptual.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Usuario** | | | | | | | |
| **login** | | **tipoID** | | **numeroID** | | **relacionU** | |
| varchar | | varchar | | number | | varchar | |
| PK, UA, NN, ND, NC | | NN, CK(CC, TI, PA) | | NN, ND, UA | | NN | |
| **Operador** | | | | | | | |
| **id** | **numeroRNT** | | **vencimientoRNT** | **registroSuperTurismo** | **vencimientoRegistroST** | | **categoria** |
| Varchar | number | | date | varchar | date | | varchar |
| PK, FK(Usuario.login) | NULL, ND, NC, UA | | NULL, UA | NULL, UA | NULL, UA | | NN, UA, CK(HOTEL, HOSTAL, P\_NATURAL, APARTAMENTO, VECINOS, VIVIENDA\_U) |
| **Operador** | | | | | | | |
| **direccion** | **horaApertura** | | **horaCierre** | **tiempoMinimo** | **gananciaAnioActual** | | **gananciaAnioCorrido** |
| varchar | date | | date | number | number | | number |
| NN, UA | NN, UA | | NN, UA | NN, UA, CK(>0) | NN, SA | | NN, SA |

|  |  |
| --- | --- |
| **Cliente** | |
| **id** | **medioPago** |
| Varchar | varchar |
| PK, FK(Usuario.login) | NN, UA |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Propiedad** | | | | | | | | |
| **id** | **capacidad** | **precio** | **tamanio** | **diasReservados** | **piso** | **habilitada** | **FechaCreacion** | **Operador** |
| number | number | number | number | number | number | number | Date | Varchar |
| PK, NN, ND, NC, SA | NN, UA, CK(>0) | NN, UA, CK(>=0) | NN, UA, CK(>0) | NN, SA | NN, UA | NN, CK(0,1) | NN | FK(Operador.id) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Habitacion** | | | | |
| **id** | **individual** | **esquema** | **Operador** | **Tipo** |
| number | boolean | varchar | Varchar | Varchar |
| FK(id.Propiedad) |  |  | FK(usuario.login) | NN |
| PK, NN, ND, NC, SA | NN, UA | NN, UA |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Apartamento** | | | | | | |
| **id** | **amueblado** | **habitaciones** | **descripcionMenaje** | **descripcionSeguro** | **tieneSeguro** | **Operador** |
| number | boolean | number | varchar | varchar | boolean | Varchar |
| FK(id.Propiedad) |  |  |  |  |  | FK(usuario.login) |
| PK, NN, ND, NC, SA | NN, UA | NN, UA, CK(>0) | NN, UA | NULL | NN, UA |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Servicio** | | | | |
| **id** | **tipo** | **precio** | **intervaloDePago** | **idPropiedad** |
| number | vachar | number | number | number |
| PK, NN, ND, NC, SA | NN, ND, UA, NC | NN, UA, CK(>=0) | NULL | FK(id.Propiedad) |
|  |  |  |  | NN |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Reserva** | | | | | | | | |
| **id** | **fechaInicio** | **fechaFin** | **personas** | **finCancelacionOportuna** | **porcentajeAPagar** | **montoTotal** | **Propiedad** | **Colectiva** |
| number | date | date | number | date | number | number | number | number |
| PK, NN, ND, NC, SA | NN, UA, NC | NN, UA, NC | NN, UA, CK(>0) | NN, SA, NC | NN, SA | NN, SA | FK(id.Propiedad) | FK(id.ReservaColectiva) |
|  |  |  |  |  |  |  | NN, NC, UA | NC |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ReservaColectiva** | | | | | |
| **id** | **tipo** | **cantidad** | **fechaInicio** | **duracion** | **Cliente** |
| number | varchar | number | date | number | Varchar |
| PK, NN, ND, NC, SA | NN, UA | NN, UA, CK(>0) | NN, UA, NC | NN, SA,CK(>0) | PK, FK(Usuario.login) |

|  |  |
| --- | --- |
| **ReservaApartemento** | |
| **idApartamento** | **idReserva** |
| number | number |
| PK | PK |
| FK(Apartamento.Propiedad.id) | FK(Reserva.id) |

|  |  |
| --- | --- |
| **ReservaHabitacion** | |
| **idHabitacion** | **idReserva** |
| number | number |
| PK | PK |
| FK(Habitacion.Propiedad.id) | FK(Reserva.id) |

1. **Diseño y construcción de la Base de Datos**
   1. **Sentencias para la creación de las tablas**

--- Sentencias SQL para la creación del esquema de parranderos

--- Las tablas tienen prefijo A\_ para facilitar su acceso desde SQL Developer

-- USO

-- Copie el contenido de este archivo en una pestaña SQL de SQL Developer

-- Ejecútelo como un script - Utilice el botón correspondiente de la pestaña utilizada

-- Creación del secuenciador

create sequence alohandes\_sequence;

-- Creación de la tabla USUARIO y especificación de sus restricciones

CREATE TABLE A\_USUARIO

(ID NUMBER,

TIPOID VARCHAR2(255 BYTE),

LOGIN VARCHAR2(255 BYTE),

RELACIONU VARCHAR2(255 BYTE),

CONSTRAINT A\_USUARIO\_PK PRIMARY KEY (LOGIN));

ALTER TABLE A\_USUARIO

ADD CONSTRAINT UN\_ID

UNIQUE (ID)

ENABLE;

ALTER TABLE A\_USUARIO

ADD CONSTRAINT CS\_U\_TIPOID

CHECK (TIPOID IN ('CARNET\_U', 'CEDULA', 'PASAPORTE'))

ENABLE;

-- Creación de la tabla OPERADOR y especificación de sus restricciones

CREATE TABLE A\_OPERADOR

(ID VARCHAR(255 BYTE),

NUMERO\_RNT NUMBER,

VENCIMIENTO\_RNT DATE,

REGISTRO\_SUPER\_TURISMO VARCHAR(255 BYTE),

VENCIMIENTO\_REGISTRO\_ST DATE,

CATEGORIA VARCHAR(255 BYTE),

DIRECCION VARCHAR(511 BYTE),

HORA\_APERTURA DATE,

HORA\_CIERRE DATE,

TIEMPO\_MINIMO NUMBER,

GANANCIA\_ANIO\_ACTUAL NUMBER,

GANANCIA\_ANIO\_CORRIDO NUMBER,

CONSTRAINT A\_OPERADOR\_PK PRIMARY KEY (ID));

ALTER TABLE A\_OPERADOR

ADD CONSTRAINT FK\_V\_USUARIO

FOREIGN KEY (ID)

REFERENCES A\_USUARIO(LOGIN)

ENABLE;

ALTER TABLE A\_OPERADOR

ADD CONSTRAINT UN\_NUMERO\_RNT

UNIQUE (NUMERO\_RNT)

ENABLE;

ALTER TABLE A\_OPERADOR

ADD CONSTRAINT CK\_CATEGORIA

CHECK (CATEGORIA IN ('HOTEL', 'HOSTAL', 'P\_NATURAL', 'APARTAMENTO', 'VECINOS', 'VIVIENDA\_U'))

ENABLE;

ALTER TABLE A\_OPERADOR

ADD CONSTRAINT CS\_TIEMPO\_MINIMO

CHECK (TIEMPO\_MINIMO > 0)

ENABLE;

-- Creación de la tabla CLIENTE y especificación de sus restricciones

CREATE TABLE A\_CLIENTE

(ID VARCHAR(255 BYTE),

MEDIO\_PAGO VARCHAR(255 BYTES),

CONSTRAINT A\_CLIENTE\_PK PRIMARY KEY (ID));

ALTER TABLE A\_OPERADOR

ADD CONSTRAINT FK\_V\_USUARIO

FOREIGN KEY (LOGIN)

REFERENCES A\_USUARIO(LOGIN)

ENABLE;

ALTER TABLE A\_CLIENTE

ADD CONSTRAINT UN\_ID

UNIQUE (ID)

ENABLE;

-- Creación de la tabla PROPIEDAD y especificación de sus restricciones

CREATE TABLE A\_PROPIEDAD

(ID NUMBER,

CAPACIDAD NUMBER,

PRECIO NUMBER,

TAMANIO NUMBER,

DIAS\_RESERVADOS NUMBER,

FECHA\_CREACION DATE,

PISO NUMBER,

HABILITADA NUMBER(1,0),

OPERADOR VARCHAR(255 BYTE)

CONSTRAINT A\_PROPIEDAD\_PK PRIMARY KEY (ID)

);

ALTER TABLE A\_PROPIEDAD

ADD CONSTRAINT FK\_V\_USUARIO

FOREIGN KEY (OPERADOR)

REFERENCES A\_USUARIO(LOGIN)

ENABLE;

ALTER TABLE A\_PROPIEDAD

ADD CONSTRAINT UN\_ID

UNIQUE (ID)

ENABLE;

ALTER TABLE A\_PROPIEDAD

ADD CONSTRAINT CK\_CAPACIDAD

CHECK (CAPACIDAD > 0)

ENABLE;

ALTER TABLE A\_PROPIEDAD

ADD CONSTRAINT CK\_PRECIO

CHECK (PRECIO > 0)

ENABLE;

ALTER TABLE A\_PROPIEDAD

ADD CONSTRAINT CK\_TAMANIO

CHECK (TAMANIO > 0)

ENABLE;

-- Creación de la tabla HABITACION y especificación de sus restricciones

CREATE TABLE A\_HABITACION

(ID NUMBER,

TIPO NUMBER(0,1,2),

INDIVIDUAL NUMBER(1,0),

ESQUEMA VARCHAR2(250 BYTE),

OPERADOR VARCHAR()255 BYTE),

CONSTRAINT A\_HABITACION\_PK PRIMARY KEY (ID)

);

ALTER TABLE A\_HABITACION

ADD CONSTRAINT FK\_V\_USUARIO

FOREIGN KEY (OPERADOR)

REFERENCES A\_USUARIO(LOGIN)

ENABLE;

ALTER TABLE A\_HABITACION

ADD CONSTRAINT FK\_V\_PROPIEDAD

FOREIGN KEY (ID)

REFERENCES A\_PROPIEDAD(ID)

ENABLE;

ALTER TABLE A\_HABITACION

ADD CONSTRAINT UN\_ID

UNIQUE (ID)

ENABLE;

-- Creación de la tabla APARTAMENTO y especificación de sus restricciones

CREATE TABLE "A\_APARTAMENTO"

(ID NUMBER,

AMUEBLADO NUMBER(1,0),

HABITACIONES NUMBER,

DESCRIPCION\_MENAJE VARCHAR2(250 BYTE),

DESCRIPCION\_SEGURO VARCHAR2(250 BYTE),

TIENE\_SEGURO NUMBER(1,0),

OPERADOR VARCHAR()255 BYTE),

CONSTRAINT A\_APARTAMENTO\_PK PRIMARY KEY (ID)

);

ALTER TABLE A\_APARTAMENTO

ADD CONSTRAINT FK\_V\_USUARIO

FOREIGN KEY (OPERADOR)

REFERENCES A\_USUARIO(LOGIN)

ENABLE;

ALTER TABLE A\_APARTAMENTO

ADD CONSTRAINT FK\_V\_PROPIEDAD

FOREIGN KEY (ID)

REFERENCES A\_PROPIEDAD(ID)

ENABLE;

ALTER TABLE A\_APARTAMENTO

ADD CONSTRAINT UN\_ID

UNIQUE (ID)

ENABLE;

ALTER TABLE A\_APARTAMENTO

ADD CONSTRAINT CK\_HABITACIONES

CHECK (HABITACIONES > 0)

ENABLE;

-- Creación de la tabla SERVICIO y especificación de sus restricciones

CREATE TABLE A\_SERVICIO

(

ID NUMBER,

TIPO VARCHAR2(250 BYTE),

PRECIO NUMBER,

INTERVALO\_PAGO NUMBER,

PROPIEDAD NUMBER,

CONSTRAINT A\_SERVICIO\_PK PRIMARY KEY (ID),

);

ALTER TABLE A\_SERVICIO

ADD CONSTRAINT FK\_V\_PROPIEDAD

FOREIGN KEY (PROPIEDAD)

REFERENCES A\_PROPIEDAD(ID)

ENABLE;

ALTER TABLE A\_SERVICIO

ADD CONSTRAINT UN\_ID

UNIQUE (ID)

ENABLE;

ALTER TABLE A\_SERVICIO

ADD CONSTRAINT CK\_PRECIO

CHECK (PRECIO >= 0)

ENABLE;

-- Creaación de la tabla RESERVA COLECTIVA y especificación de sus restricciones

CREATE TABLE A\_RESERVACOLECTIVA

(ID NUMBER,

FECHA\_INICIO DATE,

DURACION NUMBER,

CANTIDAD NUMBER,

TIPO DATE,

CLIENTE VARCHAR(255 BYTE)

CONSTRAINT A\_RESERVA\_PK PRIMARY KEY (ID)

);

ALTER TABLE A\_RESERVACOLECTIVA

ADD CONSTRAINT UN\_ID

UNIQUE (ID)

ENABLE;

ALTER TABLE A\_RESERVACOLECTIVA

ADD CONSTRAINT CK\_CANTIDAD

CHECK (CANTIDAD > 0)

ENABLE;

ALTER TABLE A\_RESERVACOLECTIVA

ADD CONSTRAINT FK\_V\_CLIENTE

FOREIGN KEY (CLIENTE)

REFERENCES A\_CLIENTE(ID)

ENABLE;

-- Creaación de la tabla RESERVA y especificación de sus restricciones

CREATE TABLE A\_RESERVA

(ID NUMBER,

FECHA\_INICIO DATE,

FECHA\_FIN DATE,

PERSONAS NUMBER,

FIN\_CANCELACION\_OPORTUNA DATE,

PORCENTAJE\_A\_PAGAR NUMBER,

MONTO\_TOTAL NUMBER,

PROPIEDAD NUMBER,

COLECTIVA NUMBER,

CONSTRAINT A\_RESERVA\_PK PRIMARY KEY (ID));

ALTER TABLE A\_RESERVA

ADD CONSTRAINT UN\_ID

UNIQUE (ID)

ENABLE;

ALTER TABLE A\_RESERVA

ADD CONSTRAINT CK\_PERSONAS

CHECK (PERSONAS > 0)

ENABLE;

ALTER TABLE A\_RESERVA

ADD CONSTRAINT FK\_V\_PROPIEDAD

FOREIGN KEY (PROPIEDAD)

REFERENCES A\_PROPIEDAD(ID)

ENABLE;

ALTER TABLE A\_RESERVA

ADD CONSTRAINT FK\_V\_RESERVACOLECTIVA

FOREIGN KEY (COLECTIVA)

REFERENCES A\_RESERVACOLECTIVA(ID)

ENABLE;

-- Creaación de la tabla RESERVAAPARTAMENTO y especificación de sus restricciones

CREATE TABLE A\_RESERVAAPARTAMENTO

(ID\_APARTAMENTO NUMBER,

ID\_RESERVA NUMBER,

CONSTRAINT A\_RESERVA\_PK PRIMARY KEY (ID\_APARTAMENTO, ID\_RESERVA)

);

ALTER TABLE A\_RESERVAAPARTAMENTO

ADD CONSTRAINT FK\_V\_APARTAMENTO

FOREIGN KEY (ID\_APARTAMENTO)

REFERENCES A\_APARTAMENTO(ID)

ENABLE;

ALTER TABLE A\_RESERVAAPARTAMENTO

ADD CONSTRAINT FK\_V\_RESERVA

FOREIGN KEY (ID\_RESERVA)

REFERENCES A\_RESERVA(ID)

ENABLE;

-- Creación de la tabla RESERVAHABITACION y especificación de sus restricciones

CREATE TABLE A\_RESERVAHABITACION

(

ID\_HABITACION NUMBER,

ID\_RESERVA NUMBER,

CONSTRAINT A\_RESERVA\_PK PRIMARY KEY (ID\_HABITACION, ID\_RESERVA)

);

ALTER TABLE A\_RESERVAHABITACION

ADD CONSTRAINT FK\_V\_HABITACION

FOREIGN KEY (ID\_HABITACION)

REFERENCES A\_HABITACION(ID)

ENABLE;

ALTER TABLE A\_RESERVAHABITACION

ADD CONSTRAINT FK\_V\_RESERVA

FOREIGN KEY (ID\_RESERVA)

REFERENCES A\_RESERVA(ID)

ENABLE;

**Requerimientos**

**Requerimiento RFC10: Consultar consumo en Alohandes**

Para mantener la privacidad se pregunta primero el login del usuario, y luego el id de la propiedad, se busca si la propiedad pertenece al usuario con la siguiente sentencia:

SELECT \*

FROM A\_Operador o NATURAL INNER JOIN A\_PROPIEDAD p

WHERE o.id = #pIdOp

ORDERED BY #pOrden

Si lo anterior retorna null es porque no pertenece asi que no se da más información

Si la propiedad pertenece al usuario que consulta entonces hay 2 posibilidades:

-- En caso de que la reserva sea en un

SELECT count(c.reservas) as res,\*

FROM A\_Cliente c NATURAL INNER JOIN A\_PROPIEDAD p, A\_HABITACION h, A\_APARTAMENTO a,

A\_RESERVAHABITACION rh, A\_RESESRVAAPARTAMENTO ra

WHERE p.id = #pIdProp AND r.FechaInicio BETWEEN TO\_DATE(#pFI) AND TO\_DATE (#pFF) AND res>0

ORDERED BY #pOrden

**Requerimiento RFC11: Consultar consumo en Alohandes v2**

Se verifica que la propiedad le pertenezca al usuario de la misma forma que en el RFC10

-- En caso que la reserva sea en un

SELECT count(c.reservas) as res,\*

FROM A\_Cliente c NATURAL INNER JOIN A\_PROPIEDAD p, A\_HABITACION h, A\_APARTAMENTO a,

A\_RESERVAHABITACION rh, A\_RESESRVAAPARTAMENTO ra

WHERE p.id = #pIdProp AND r.FechaInicio BETWEEN TO\_DATE(#pFI) AND TO\_DATE (#pFF) AND res=0

ORDERED BY #pOrden

**Requerimiento RFC12: Consultar funcionaminento**

Se divide la consulta en las siguientes:

Para obtener

with semana as (

SELECT to\_number(to\_char(to\_date(r.fecha\_inicio,'MM/DD/YYYY'),'WW')) week\_num , MAX(p.dias\_reservados) AS personas, p.id as id

FROM A\_RESERVA r

INNER JOIN A\_PROPIEDAD p ON (p.id = r.propiedad)

group by to\_number(to\_char(to\_date(r.fecha\_inicio,'MM/DD/YYYY'),'WW')),p.id

)

SELECT p.\*

FROM A\_PROPIEDAD p

INNER JOIN semana s ON (s.id = p.id)

Para obtener

with semana as (

SELECT to\_number(to\_char(to\_date(r.fecha\_inicio,'MM/DD/YYYY'),'WW')) week\_num , MIN(p.dias\_reservados) AS personas, p.id as id

FROM A\_RESERVA r

INNER JOIN A\_PROPIEDAD p ON (p.id = r.propiedad)

group by to\_number(to\_char(to\_date(r.fecha\_inicio,'MM/DD/YYYY'),'WW')),p.id

)

SELECT p.\*

FROM A\_PROPIEDAD p

INNER JOIN semana s ON (s.id = p.id)

Para obtener

with semana as (

SELECT to\_number(to\_char(to\_date(r.fecha\_inicio,'MM/DD/YYYY'),'WW')) week\_num , COUNT(r.propiedad) AS res, o.id as id

FROM A\_RESERVA r

INNER JOIN A\_PROPIEDAD p ON (p.id = r.propiedad)

INNER JOIN A\_OPERADOR o ON (o.id = p.operador)

group by to\_number(to\_char(to\_date(r.fecha\_inicio,'MM/DD/YYYY'),'WW')),o.id

)

SELECT MAX(s.res), o.id, o.numero\_rnt, o.vencimiento\_rnt, o.registro\_super\_turismo, o.vencimiento\_registro\_st, o.categoria, o.direccion, o.hora\_apertura,o.hora\_cierre

,o.tiempo\_minimo, o.ganancia\_anio\_actual, o.ganancia\_anio\_corrido, o.habitacion, o.apartamento

FROM A\_OPERADOR o

INNER JOIN semana s ON (s.id = o.id) group by o.id, o.numero\_rnt, o.vencimiento\_rnt, o.registro\_super\_turismo, o.vencimiento\_registro\_st,

o.categoria, o.direccion, o.hora\_apertura, o.hora\_cierre, o.tiempo\_minimo,

o.ganancia\_anio\_actual, o.ganancia\_anio\_corrido, o.habitacion, o.apartamento

Finalmente:

with semana as (

SELECT to\_number(to\_char(to\_date(r.fecha\_inicio,'MM/DD/YYYY'),'WW')) week\_num , COUNT(r.propiedad) AS res, o.id as id

FROM A\_RESERVA r

INNER JOIN A\_PROPIEDAD p ON (p.id = r.propiedad)

INNER JOIN A\_OPERADOR o ON (o.id = p.operador)

group by to\_number(to\_char(to\_date(r.fecha\_inicio,'MM/DD/YYYY'),'WW')),o.id

)

SELECT MIN(s.res), o.id, o.numero\_rnt, o.vencimiento\_rnt, o.registro\_super\_turismo, o.vencimiento\_registro\_st, o.categoria, o.direccion, o.hora\_apertura,o.hora\_cierre

,o.tiempo\_minimo, o.ganancia\_anio\_actual, o.ganancia\_anio\_corrido, o.habitacion, o.apartamento

FROM A\_OPERADOR o

INNER JOIN semana s ON (s.id = o.id) group by o.id, o.numero\_rnt, o.vencimiento\_rnt, o.registro\_super\_turismo, o.vencimiento\_registro\_st,

o.categoria, o.direccion, o.hora\_apertura, o.hora\_cierre, o.tiempo\_minimo,

o.ganancia\_anio\_actual, o.ganancia\_anio\_corrido, o.habitacion, o.apartamento

**Requerimiento RFC13: Consultar buenos clientes**

with res as ( SELECT count(\*) AS mes , c.id as id, count(c.reservas) as reservas

FROM A\_CLIENTE c

INNER JOIN A\_RESERVA r ON (c.reservas = r.id)

INNER JOIN A\_PROPIEDAD p ON (p.id = r.propiedad)

WHERE to\_char(r.fecha\_inicio,'MM')- to\_char((

SELECT r2.fecha\_inicio

FROM A\_CLIENTE c2

INNER JOIN A\_RESERVA r2 ON (c2.reservas = r2.id)

INNER JOIN A\_PROPIEDAD p2 ON (p2.id = r2.propiedad)

WHERE c.id = c2.id)) = -1

group by c.id

)

SELECT c.\*, p.precio , h.tipo, re.mes

FROM A\_CLIENTE c

INNER JOIN A\_RESERVA r ON (c.reservas = r.id)

INNER JOIN A\_PROPIEDAD p ON (p.id = r.propiedad)

INNER JOIN A\_HABITACION h ON (p.id = h.id)

INNER JOIN A\_APARTAMENTO ap ON (p.id = ap.id)

INNER JOIN res re ON (re.id = c.id)

WHERE p.precio > 150 OR h.tipo = 3 OR re.mes = re.reservas