

## Control N°1: Definición y Manipulación SQL

Valeria Beratto Ulloa

Nombre:

RUT:

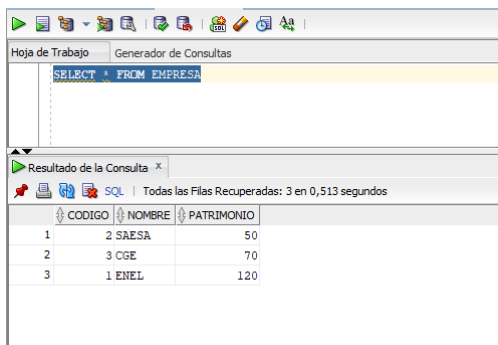
Dado el siguiente Modelo Relacional, genere las siguientes operaciones de DDL y DML (100 puntos).

Factura ( **fac\_cod**, fac\_fecha)  
Proveedor (**pro\_rut**, pro\_nombre, pro\_representante)  
Cliente (**cli\_rut**, cli\_nombre)  
Producto (**prd\_cod**, prd\_nombre, prd\_precio, *pro\_rut*)  
Compra (*fac\_cod*, *cli\_rut*, *prd\_cod*, com\_cantidad)

Los atributos en negrita son claves primarias y los cursivos son claves foráneas.

### Indicaciones Generales:

1. Para cada pregunta, que se detallan a continuación, debe colocar una fotografía o screenshot que contenga del código (SQL) y resultado obtenido en ORACLE SQL Developer u ORACLE SQL LIVE. Pregunta que no contenga imagen **no será evaluada**.



2. Realizar al menos dos consultas con el operador join
3. **Tiempo límite para subir el test a la plataforma en formato pdf, miércoles 19 de octubre 23:59 hrs.**
4. Cualquier inconveniente enviar un correo electrónico a [vberatto@ubiobio.cl](mailto:vberatto@ubiobio.cl)

Resuelva:

## 1. Crear en SQL cada una de las tablas del esquema anterior (10 puntos)

Hoja de Trabajo

Generador de Consultas

```

2  --1. Crear en SQL cada una de las tablas del esquema anterior
3  --Factura ( fac_cod, fac_fecha)
4  CREATE TABLE T1_FACTURA(
5      FAC_COD INTEGER,
6      FAC_FECHA DATE NOT NULL,
7      PRIMARY KEY(FAC_COD)
8  );
9  --Proveedor (pro_rut, pro_nombre, pro_representante)
10 CREATE TABLE T1_PROVEEDOR(
11     PRO_RUT VARCHAR(11),
12     PRO_NOMBRE VARCHAR(50) NOT NULL,
13     PRO_REPRESENTANTE VARCHAR(30) NOT NULL,
14     PRIMARY KEY(PRO_RUT)
15 );
16 --Cliente (cli_rut, cli_nombre)
17 CREATE TABLE T1_CLIENTE(
18     CLI_RUT VARCHAR(11),
19     CLI_NOMBRE VARCHAR(50) NOT NULL,
20     PRIMARY KEY(CLI_RUT)
21 );
22 --Producto (prd_cod, prd_nombre, prd_precio, pro_rut)
23 CREATE TABLE T1_PRODUCTO(
24     PRD_COD INTEGER,
25     PRD_NOMBRE VARCHAR(50) NOT NULL,
26     PRD_PRECIO INTEGER NOT NULL,
27     PRO_RUT VARCHAR(11) NOT NULL,
28     PRIMARY KEY (PRD_COD),
29     FOREIGN KEY(PRO_RUT) REFERENCES T1_PROVEEDOR(PRO_RUT)
30 );
31 --Compra (fac_cod, cli_rut, prd_cod, com_cantidad)

```

```

31 --Compra (fac_cod, cli_rut, prd_cod, com_cantidad)
32 CREATE TABLE T1_COMPRA(
33     FAC_COD INTEGER,
34     CLI_RUT VARCHAR(11),
35     PRD_COD INTEGER,
36     COM_CANTIDAD INTEGER NOT NULL,
37     PRIMARY KEY(FAC_COD, CLI_RUT, PRD_COD),
38     FOREIGN KEY(FAC_COD) REFERENCES T1_FACTURA(FAC_COD),
39     FOREIGN KEY(CLI_RUT) REFERENCES T1_CLIENTE(CLI_RUT),
40     FOREIGN KEY(PRD_COD) REFERENCES T1_PRODUCTO(PRD_COD)
41 );

```

Salida de Script

Tarea terminada en 1,401 segundos

Table T1\_FACTURA creado.

Table T1\_PROVEEDOR creado.

Table T1\_CLIENTE creado.

Table T1\_PRODUCTO creado.

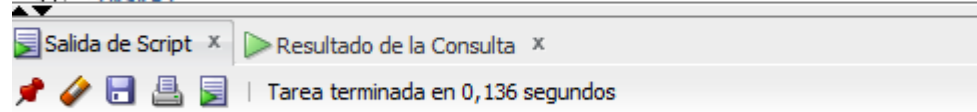
Table COMPRA creado.

2. Insertar dos registros por cada tabla (10 puntos)

```

42 |
43 | -- INSERCIONES
44 | INSERT
45 | INTO T1_FACTURA(FAC_COD, FAC_FECHA)
46 | VALUES (1, '01-10-2021');
47 | INSERT
48 | INTO T1_FACTURA(FAC_COD, FAC_FECHA)
49 | VALUES (2, '14-10-2021');
50 | INSERT
51 | INTO T1_FACTURA(FAC_COD, FAC_FECHA)
52 | VALUES (3, '15-10-2021');
53 | INSERT
54 | INTO T1_FACTURA(FAC_COD, FAC_FECHA)
55 | VALUES (4, '15-10-2021');
56 |
57 | INSERT

```



Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,136 segundos

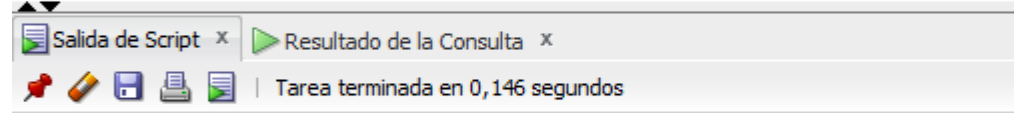
1 fila insertadas.

1 fila insertadas.

```

57 | INSERT
58 | INTO T1_PROVEEDOR(PRO_RUT, PRO_NOMBRE, PRO_REPRESENTANTE)
59 | VALUES ('17000309-8', 'JUAN LEIVA', 'EMPRESA1');
60 | INSERT
61 | INTO T1_PROVEEDOR(PRO_RUT, PRO_NOMBRE, PRO_REPRESENTANTE)
62 | VALUES ('15300309-8', 'MANUEL SOLARI', 'EMPRESA2');
63 |

```



Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,146 segundos

1 fila insertadas.

1 fila insertadas.

```

64 INSERT
65 INTO T1_CLIENTE(CLI_RUT, CLI_NOMBRE)
66 VALUES ('18001112-2', 'LAURA TAPIA');
67 INSERT
68 INTO T1_CLIENTE(CLI_RUT, CLI_NOMBRE)
69 VALUES ('20301092-3', 'RAFAEL ABARCA');
70 INSERT
71 INTO T1_CLIENTE(CLI_RUT, CLI_NOMBRE)
72 VALUES ('16272498-1', 'PABLO TORRES');
73 INSERT
74 INTO T1_CLIENTE(CLI_RUT, CLI_NOMBRE)
75 VALUES ('15000123-0', 'MARCELO CASZELY');
76

```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,113 segundos

1 fila insertadas.

1 fila insertadas.

```

77 INSERT
78 INTO T1_PRODUCTO(PRD_COD, PRD_NOMBRE, PRD_PRECIO, PRO_RUT)
79 VALUES (1, 'BICICLETA', 140000, '15300309-8');
80 INSERT
81 INTO T1_PRODUCTO(PRD_COD, PRD_NOMBRE, PRD_PRECIO, PRO_RUT)
82 VALUES (2, 'AUTO', 5000000, '17000309-8');
83 INSERT
84 INTO T1_PRODUCTO(PRD_COD, PRD_NOMBRE, PRD_PRECIO, PRO_RUT)
85 VALUES (3, 'COMPUTADOR', 500000, '15300309-8');
86 INSERT
87 INTO T1_PRODUCTO(PRD_COD, PRD_NOMBRE, PRD_PRECIO, PRO_RUT)
88 VALUES (4, 'CAMA', 120000, '17000309-8');
89
90

```

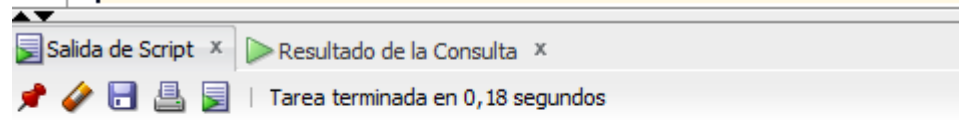
Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,115 segundos

1 fila insertadas.

1 fila insertadas.

```
90 | INSERT
91 | INTO T1_COMPRA(FAC_COD, CLI_RUT, PRD_COD, COM_CANTIDAD)
92 | VALUES (1, '18001112-2', 1, 1);
93 | INSERT
94 | INTO T1_COMPRA(FAC_COD, CLI_RUT, PRD_COD, COM_CANTIDAD)
95 | VALUES (1, '18001112-2', 4, 1);
96 | INSERT
97 | INTO T1_COMPRA(FAC_COD, CLI_RUT, PRD_COD, COM_CANTIDAD)
98 | VALUES (2, '16272498-1', 3, 1);
99 |
```



1 fila insertadas.

1 fila insertadas.

3. Muestre el nombre de los clientes que han comprado al menos dos productos ( no la cantidad que se compró) cuyo precio sea mayor a \$3.000.- (20 puntos)

```

100  --3.    Muestre el nombre de los clientes que han comprado al menos dos productos
101  --( no la cantidad que se compró) cuyo precio sea mayor a $3.000.-
102  SELECT A.CLI_NOMBRE
103  FROM T1_COMPRA C
104  INNER JOIN T1_CLIENTE A ON A.CLI_RUT = C.CLI_RUT
105  INNER JOIN T1_PRODUCTO P ON P.PR_D_COD = C.PR_D_COD
106  WHERE P.PR_D_PRECIO >3000
107  GROUP BY A.CLI_NOMBRE
108  HAVING COUNT (C.PR_D_COD) >=2;
109
110
111

```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,011 segundos

CLI_NOMBRE
1 LAURA TAPIA

4. Seleccione el nombre de los clientes que no han comprado productos (20 puntos)

```

110  --4.    Seleccione el nombre de los clientes que no han comprado productos
111  SELECT C.CLI_NOMBRE
112  FROM T1_CLIENTE C
113  LEFT JOIN T1_COMPRA A ON A.CLI_RUT = C.CLI_RUT
114  WHERE A.PR_D_COD IS NULL
115
116
117

```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 2 en 0,017 segundos

CLI_NOMBRE
1 RAFAEL ABARCA
2 MARCELO CASZELY

5. Seleccione el nombre de los clientes que han adquirido bicicletas (20 puntos)

```

116 --5.     Seleccione el nombre de los clientes que han adquirido bicicletas
117
118 SELECT A.CLI_NOMBRE
119 FROM T1_COMPRA C
120 INNER JOIN T1_CLIENTE A ON A.CLI_RUT = C.CLI_RUT
121 INNER JOIN T1_PRODUCTO P ON P.PR_D_COD = C.PR_D_COD
122 WHERE P.PR_D_NOMBRE = 'BICICLETA';
123

```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,01 segundos

CLI_NOMBRE
1 LAURA TAPIA

6. Muestre el nombre del proveedor que haya emitido la menor cantidad de facturas durante el mes de octubre. Use vistas (20 puntos)

LA SIGUIENTE VISTA CUENTA LAS FACTURAS EMITIDAS SIN CONTAR LOS NULOS

```

127 --CUENTA LOS NO NULOS
128 CREATE VIEW T1_CANTFACT_EMITIDAS_PROV AS(
129     SELECT A.PRO_RUT, COUNT(1) AS CANT_FACT
130     FROM T1_FACTURA F, T1_PRODUCTO P
131     FULL OUTER JOIN T1_PROVEEDOR A ON A.PRO_RUT = P.PRO_RUT
132     INNER JOIN T1_COMPRA C ON C.PR_D_COD = P.PR_D_COD
133     WHERE F.FAC_COD = C.FAC_COD AND F.FAC_FECHA BETWEEN '01-10-2021' AND '31-10-2021'
134     GROUP BY A.PRO_RUT
135 );
136

```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 2 en 0,004 segundos

PRO_RUT	CANT_FACT
1 17000309-8	1
2 15300309-8	2

UTILIZO LA SEGUNDA VISTA PARA CONTAR LOS VALORES NULOS

```
--AGREGA LOS NULOS
CREATE VIEW T1_CANT_BOLETAS_EMI_PROVEEDOR AS(
    SELECT P.PRO_NOMBRE,
    CASE
        WHEN CANT_FACT IS NULL THEN 0
        ELSE CANT_FACT
    END AS CANT_FACT
    FROM T1_CANTFACT_EMITIDAS_PROV T
    RIGHT JOIN T1_PROVEEDOR P ON P.PRO_RUT = T.PRO_RUT
);
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 3 en 0,054 segundos

	PRO_NOMBRE	CANT_FACT
1	JUAN LEIVA	1
2	MANUEL SOLARI	2
3	JUAN LORCA	0

CON LA SIGUIENTE CONSULTA SE OBTIENE EL NOMBRE DEL PROVEEDOR QUE TIENE LA MENOR CANTIDAD DE FACTURAS EMITIDAS

```
147
148 SELECT P.PRO_NOMBRE
149 FROM T1_CANT_BOLETAS_EMI_PROVEEDOR P
150 WHERE P.CANT_FACT = (SELECT MIN(CANT_FACT) FROM T1_CANT_BOLETAS_EMI_PROVEEDOR);
151
152
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 1 en 0,158 segundos

	PRO_NOMBRE
1	JUAN LORCA