PRÁCTICA 1:

Toma de contacto con el SGBD Oracle y consultas SQL

Ficheros y Bases de Datos

Javier Martín Moreno-Manzanaro

3º I.T.I.G.

**Apartado 1: crea\_tablas.sql**

CREATE TABLE Restaurantes

( código Number(8) NOT NULL,

nombre Char(20),

calle Char(30),

"código postal" Char(5),

comisión Number(8,2),

PRIMARY KEY (código)

);

CREATE SEQUENCE restaurantes\_seq;

INSERT INTO Restaurantes VALUES (12345678, 'NombreRestaurante\_A', 'calle\_A', 'AAAAA', 123456.78);

INSERT INTO Restaurantes VALUES (01234567, 'NombreRestaurante\_B', 'calle\_B', 'BBBBB', 123456.78);

CREATE TABLE "Areas Cobertura"

( restaurante Number(8) NOT NULL,

"código postal" Char(5) NOT NULL,

PRIMARY KEY (restaurante, "código postal"),

FOREIGN KEY (restaurante) REFERENCES Restaurantes ON DELETE CASCADE

);

INSERT INTO "Areas Cobertura" VALUES (12345678, 'XXXXX');

INSERT INTO "Areas Cobertura" VALUES (01234567, 'YYYYY');

CREATE TABLE Horarios

( restaurante Number(8) NOT NULL,

"día semana" Char (1) NOT NULL,

hora\_apertura Date,

hora\_cierre Date,

PRIMARY KEY (restaurante, "día semana"),

FOREIGN KEY (restaurante) REFERENCES Restaurantes ON DELETE CASCADE

);

INSERT INTO Horarios VALUES (12345678, 'L', TO\_DATE('09:00', 'HH24:MI'), TO\_DATE('22:00', 'HH24:MI'));

INSERT INTO Horarios VALUES (01234567, 'D', TO\_DATE('09:00', 'HH24:MI'), TO\_DATE('22:00', 'HH24:MI'));

CREATE TABLE Platos

( restaurante Number(8) NOT NULL,

nombre Char(20) NOT NULL,

precio Number(8,2),

descripción Char(30),

categoría Char(10),

PRIMARY KEY (restaurante, nombre),

FOREIGN KEY (restaurante) REFERENCES Restaurantes ON DELETE CASCADE

);

CREATE INDEX indice\_platos ON Platos (categoría);

INSERT INTO Platos VALUES (12345678, 'NombrePlato\_C', 123456.78, 'filete de ternera', 'carnes');

INSERT INTO Platos VALUES (01234567, 'NombrePlato\_D', 123456.78, 'merluza a la vasca', 'pescados');

CREATE TABLE Clientes

( DNI Char(9) NOT NULL,

nombre Char (20),

apellidos Char (20),

calle Char (20),

número Number(4),

piso Char(5),

localidad Char(15),

"código postal" Char(5),

teléfono Char (9),

usuario Char (8) UNIQUE,

contraseña Char(8) DEFAULT ‘Nopass’,

PRIMARY KEY (DNI)

);

INSERT INTO Clientes VALUES ('98765432K', 'Manuel', 'Fernández Gómez', 'Gran Vía', 16, 2, 'Madrid', 'AAAAA', '600606060', 'user1', 'pass1');

INSERT INTO Clientes VALUES ('98765432W', 'Clara', 'García Nuñez', 'Capitán Almanzor', 84, 5, 'Teruel', 'BBBBB', '611616161', 'user2', 'pass2');

CREATE TABLE Pedidos

( código Number(8) NOT NULL,

estado Char(9) DEFAULT 'REST',

CONSTRAINT estado CHECK (estado IN ('REST', 'CANCEL', 'RUTA', 'ENTREGADO', 'RECHAZADO')),

fecha\_hora\_pedido Date,

fecha\_hora\_entrega Date,

"importe total" Number(8,2),

cliente Char (9),

PRIMARY KEY (código),

FOREIGN KEY (cliente) REFERENCES Clientes ON DELETE CASCADE

);

CREATE INDEX indice\_pedidos ON Pedidos (estado);

CREATE SEQUENCE pedidos\_seq START WITH 1;

INSERT INTO Pedidos VALUES (12345678, 'RECHAZADO', TO\_DATE('11-11-2011 11:11:11', 'DD-MM-YYYY HH24:MI:SS'), TO\_DATE('12-12-2011 12:12:12', 'DD-MM-YYYY HH24:MI:SS'), 123456.78, '98765432K');

INSERT INTO Pedidos VALUES (01234567, 'CANCEL', TO\_DATE('12-12-2011 12:12:12', 'DD-MM-YYYY HH24:MI:SS'), TO\_DATE('11-11-2011 11:11:11', 'DD-MM-YYYY HH24:MI:SS'), 122223.78, '98765432W');

CREATE TABLE Contiene

( restaurante Number(8) NOT NULL,

plato Char(20) NOT NULL,

pedido Number(8) NOT NULL,

"precio con comisión" Number(8,2),

unidades Number(4),

PRIMARY KEY (restaurante, plato, pedido),

FOREIGN KEY (pedido) REFERENCES Pedidos ON DELETE CASCADE,

FOREIGN KEY (restaurante, plato) REFERENCES Platos ON DELETE CASCADE

);

INSERT INTO Contiene VALUES (12345678, 'NombrePlato\_C', 01234567, 123456.78, 23);

INSERT INTO Contiene VALUES (01234567, 'NombrePlato\_D', 12345678, 123456.78, 5);

CREATE TABLE Descuentos

( código Number(8) NOT NULL,

fecha\_caducidad Date,

"porcentaje descuento" Number(3) CHECK (("porcentaje descuento" > 0) AND ("porcentaje descuento" <= 100)),

PRIMARY KEY (código)

);

CREATE SEQUENCE descuentos\_seq;

INSERT INTO Descuentos VALUES (12345678, '11-11-2011', 15);

INSERT INTO Descuentos VALUES (01234567, '12-12-2011', 18);

CREATE TABLE AplicadoA

( descuento Number(8) NOT NULL,

Pedido Number(8) NOT NULL,

PRIMARY KEY (descuento, pedido),

FOREIGN KEY (descuento) REFERENCES Descuentos ON DELETE CASCADE,

FOREIGN KEY (pedido) REFERENCES Pedidos ON DELETE CASCADE

);

INSERT INTO AplicadoA VALUES (01234567, 12345678);

INSERT INTO AplicadoA VALUES (12345678, 01234567);

**Apartado 2: Carga de datos con SQL Loader**

***● Archivos de Control:***

*- Control\_apli.txt*

LOAD DATA

INFILE aplicadosA.txt

INTO TABLE AplicadoA

APPEND

FIELDS TERMINATED BY ';'

(descuento, pedido)

*- Control\_areas.txt*

LOAD DATA

INFILE areas.txt

INTO TABLE "Areas Cobertura"

APPEND

FIELDS TERMINATED BY ';'

(restaurante, "código postal")

*- Control\_clientes.txt*

LOAD DATA

INFILE clientes.txt

INTO TABLE Clientes

APPEND

FIELDS TERMINATED BY ';'

(DNI, nombre, apellidos, calle, número, piso, localidad, "código postal", teléfono, usuario, contraseña "DEFAULT")

*- Control\_contiene.txt*

LOAD DATA

INFILE contiene.txt

INTO TABLE Contiene

APPEND

FIELDS TERMINATED BY ';'

(restaurante, plato, pedido, "precio con comisión", unidades)

*- Control\_descuentos.txt*

LOAD DATA

INFILE descuentos.txt

INTO TABLE Descuentos

APPEND

FIELDS TERMINATED BY ';'

(código, fecha\_caducidad DATE 'DD/MM/YYYY', "porcentaje descuento")

*- Control\_horarios.txt*

LOAD DATA

INFILE horarios.txt

INTO TABLE Horarios

APPEND

FIELDS TERMINATED BY ';'

(restaurante, "día semana", hora\_apertura DATE 'HH24:MI:SS', hora\_cierre DATE 'HH24:MI:SS')

*- Control\_pedidos.txt*

LOAD DATA

INFILE pedidos.txt

INTO TABLE Pedidos

APPEND

FIELDS TERMINATED BY ';'

(código "pedidos\_seq.NEXTVAL", estado, fecha\_hora\_pedido DATE 'DD/MM/YYYY HH24:MI:SS', fecha\_hora\_entrega DATE 'DD/MM/YYYY HH24:MI:SS', "importe total", cliente)

*- Control\_platos.txt*

LOAD DATA

INFILE platos.txt

INTO TABLE Platos

APPEND

FIELDS TERMINATED BY ';'

(restaurante, nombre, precio, descripción, categoría)

*- Control\_rest.txt*

LOAD DATA

INFILE restaurantes.txt

INTO TABLE Restaurantes

APPEND

FIELDS TERMINATED BY ';'

(código, nombre, calle, "código postal", comisión)

***● Errores:***

Registro 3: Rechazado - Error en tabla **APLICADOA**.

ORA-02291: restricción de integridad (GESB03.SYS\_C0023607) violada - clave principal no encontrada

* Este error es debido a que se trata de aplicar un descuento a un pedido que no existe (el 10) debido a que se usa una secuencia que comienza en 1 para identificar los pedidos y sólo tenemos un total de 9 (más las dos inserciones de prueba). Borramos la fila que da error del fichero de datos.

Registro 6: Rechazado - Error en tabla **"Areas Cobertura".**

ORA-02291: restricción de integridad (GESB03.SYS\_C0025032) violada - clave principal no encontrada

* Este error ocurre porque en el registro mencionado, figura originalmente el código de restaurante 5677, sin embargo no existe ningún restaurante con dicho código, mientras que sí existe el restaurante con código 5678. Solucionamos este error modificando el fichero de datos para cambiar dicho código y que la carga resulte correcta.

Registro 1: Rechazado - Error en tabla **CLIENTES**, columna CONTRASEÑA.

Columna no encontrada antes del fin de registro lógico (utilice TRAILING NULLCOLS)

Registro 2: Rechazado - Error en tabla **CLIENTES**, columna USUARIO.

Columna no encontrada antes del fin de registro lógico (utilice TRAILING NULLCOLS)

* Estos errores son debidos a que a los dos primeros usuarios les falta los campos de contraseña al primero y usuario y contraseña al segundo. Lo solucionamos poniendo en el segundo caso un usuario a mano en el fichero de datos “userEX”, y añadiendo en el fichero de control el valor DEFAULT para el campo de la contraseña. (Véase el fichero control\_clientes.txt, en la sección anterior).

Registro 5: Rechazado - Error en tabla **DESCUENTOS**.

ORA-02290: restricción de control (GESB03.SYS\_C0023601) violada

* Este error nos indica la violación de la restricción CHECK que impusimos al crear la tabla de descuentos. El porcentaje de descuento debe ser mayor que 0 y menor o igual que 100. Como figura un porcentaje del 110 por ciento el descuento no es válido y solucionamos el error borrando esa línea en el fichero de datos.

Registro 4: Rechazado - Error en tabla **HORARIOS**.

ORA-00001: restricción única (GESB03.SYS\_C0023659) violada

Registro 17: Rechazado - Error en tabla **HORARIOS**.

ORA-02291: restricción de integridad (GESB03.SYS\_C0023660) violada - clave principal no encontrada

* El error del registro 4 es debido a que se le asignan dos horarios al mismo restaurante y el mismo día de la semana, pero esto no es correcto, desde el punto de vista del enunciado, y por tanto tampoco cumple las restricciones de claves especificadas en el apartado 1. Solucionamos este error eliminando la correspondiente línea del fichero de datos.
* Por otra parte, el error del registro 17 se debe, de nuevo, a la utilización errónea del código del restaurante, 5677. Sustituimos dicho código por el 5678 para que la carga sea correcta.

Registro 5: Rechazado - Error en tabla **PLATOS**, columna CATEGORÍA.

ORA-12899: el valor es demasiado grande para la columna "GESB03"."PLATOS"."CATEGORÍA" (real: 11, máximo: 10)

Registro 7: Rechazado - Error en tabla **PLATOS**, columna CATEGORÍA.

ORA-12899: el valor es demasiado grande para la columna "GESB03"."PLATOS"."CATEGORÍA" (real: 11, máximo: 10)

Registro 9: Rechazado - Error en tabla **PLATOS**, columna NOMBRE.

ORA-12899: el valor es demasiado grande para la columna "GESB03"."PLATOS"."NOMBRE" (real: 23, máximo: 20)

* Estos errores son debidos al exceso de caracteres utilizados en los campos indicados. Como se indica en el enunciado, sustituimos la categoría “Vegetariano” por “Vegetal”, y el nombre del plato “torta de carne especial”, por “torta carne esp”, para que la carga sea correcta.

**NOTA:** Se presentan los errores mencionados en el enunciado de la práctica pero no la totalidad de los errores producidos durante la realización de la práctica, derivados de errores de transcripción o similares.

**Apartado 3: consultas.sql**

*1. Listado con todos los datos de los clientes, ordenados por apellidos.*

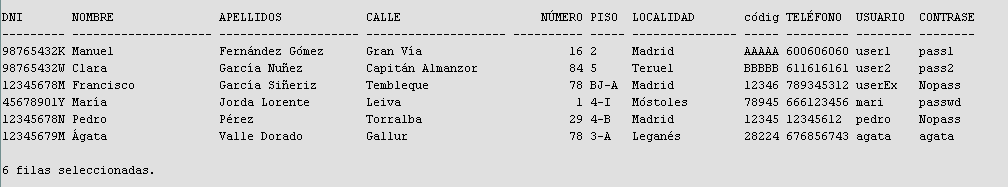
**● Consulta:**

SELECT DNI, nombre, apellidos, calle, número, piso, localidad, "código postal", teléfono, usuario, contraseña

FROM Clientes

ORDER BY apellidos

**● Salida:**

****

*2. Horarios de cada uno de los restaurantes. Para cada restaurante aparecerá el día de la semana (sustituyendo la letra por el nombre completo del día) y la hora de apertura y de cierre, en formato HH:MM*

**● Consulta:**

SELECT restaurante, ('Lunes') AS Dia, (TO\_CHAR(hora\_apertura, 'HH24:MI')) AS Hora\_apertura, (TO\_CHAR(hora\_cierre, 'HH24:MI')) AS Hora\_Cierre

FROM Restaurantes R, Horarios H

WHERE R.código = H.restaurante

AND "día semana" = 'L'

UNION

SELECT restaurante, 'Martes', TO\_CHAR(hora\_apertura, 'HH24:MI'), TO\_CHAR(hora\_cierre, 'HH24:MI')

FROM Restaurantes R, Horarios H

WHERE R.código = H.restaurante

AND "día semana" = 'M'

UNION

SELECT restaurante, 'Miercoles', TO\_CHAR(hora\_apertura, 'HH24:MI'), TO\_CHAR(hora\_cierre, 'HH24:MI')

FROM Restaurantes R, Horarios H

WHERE R.código = H.restaurante

AND "día semana" = 'X'

UNION

SELECT restaurante, 'Jueves', TO\_CHAR(hora\_apertura, 'HH24:MI'), TO\_CHAR(hora\_cierre, 'HH24:MI')

FROM Restaurantes R, Horarios H

WHERE R.código = H.restaurante

AND "día semana" = 'J'

UNION

SELECT restaurante, 'Viernes', TO\_CHAR(hora\_apertura, 'HH24:MI'), TO\_CHAR(hora\_cierre, 'HH24:MI')

FROM Restaurantes R, Horarios H

WHERE R.código = H.restaurante

AND "día semana" = 'V'

UNION

SELECT restaurante, 'Sabado', TO\_CHAR(hora\_apertura, 'HH24:MI'), TO\_CHAR(hora\_cierre, 'HH24:MI')

FROM Restaurantes R, Horarios H

WHERE R.código = H.restaurante

AND "día semana" = 'S'

UNION

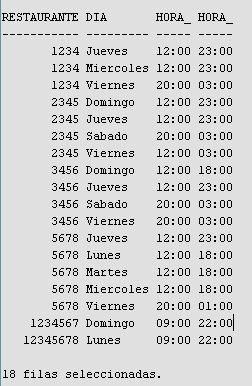
SELECT restaurante, 'Domingo', TO\_CHAR(hora\_apertura, 'HH24:MI'), TO\_CHAR(hora\_cierre, 'HH24:MI')

FROM Restaurantes R, Horarios H

WHERE R.código = H.restaurante

AND "día semana" = 'D'

**● Salida:**

**

*3. Qué clientes (DNI, nombre y apellidos) han pedido alguna vez platos de la categoría “picante”?*

**● Consulta:**

SELECT C.DNI, C.nombre, C.apellidos

FROM Clientes C, Pedidos P, Contiene Co, Platos Pl

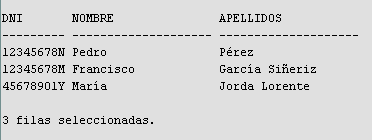
WHERE C.DNI = P.cliente

AND P.código = Co.pedido

AND Co.plato = Pl.nombre

AND Pl.categoría = 'picante'

**● Salida:**

**

*4. ¿Qué clientes (DNI, nombre y apellidos) han pedido platos en todos los restaurantes?*

**● Consulta:**

SELECT C.DNI, C.nombre, C.apellidos

FROM Clientes C, Pedidos P, Contiene Co

WHERE C.DNI = P.cliente

AND P.código = Co.pedido

GROUP BY C.DNI, C.nombre, C.apellidos

HAVING COUNT (DISTINCT Co.Restaurante) >=

(SELECT COUNT (DISTINCT código)

FROM Restaurantes

);

**● Salida:**

**

\*\*\*Si atendemos a la tabla de pedidos, veremos que el máximo nº de veces que aparece repetido un mismo DNI, es de 3 veces; y de un vistazo apreciamos que la tabla de restaurantes contiene al menos 4 restaurantes (sin contar los añadidos de prueba en el apartado 1 de la práctica), por lo que ningún cliente ha pedido platos en todos los restaurantes

*5. ¿Qué clientes (DNI, nombre y apellidos) no han recibido aún sus pedidos?*

**● Consulta:**

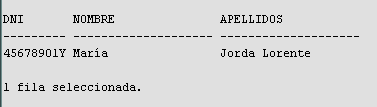
SELECT C.DNI, C.nombre, C.apellidos

FROM Clientes C, Pedidos P

WHERE C.DNI = P.cliente

AND p.estado = 'RUTA'

**● Salida:**

**

*6. Muestra todos los datos (salvo los platos que lo componen) del pedido de mayor importe total. Considera que puede haber varios pedidos con el mismo importe.*

**● Consulta:**

SELECT P.código, P.cliente, P."importe total", P.estado, (to\_char(P.fecha\_hora\_pedido, 'DD-MM-YY HH24:MI')) AS Fecha\_Hora\_Pedido, (to\_char(P.fecha\_hora\_entrega, 'DD-MM-YY HH24:MI')) AS Fecha\_Hora\_Entrega

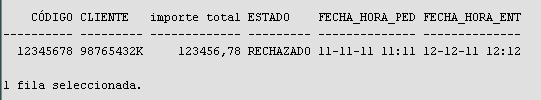
FROM Pedidos P

WHERE P."importe total" = (SELECT MAX("importe total")

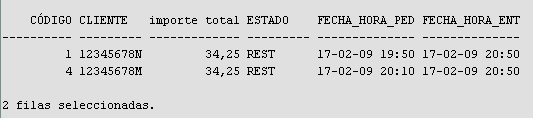
FROM Pedidos p

);

**● Salida:**

**

\*\*\*Este resultado es debido a un INSERT INTO de prueba realizado en el apartado 1. El resultado que arrojaría la misma consulta si cambiamos el importe total del cliente de prueba para que no sea el mayor, sería algo como esto:



*7. Obtén el valor medio de los pedidos de cada cliente. Muestra el DNI, nombre y apellidos de cada cliente.*

**● Consulta:**

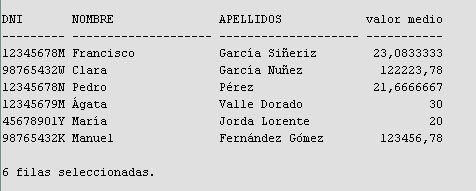
SELECT c.DNI, c.nombre, c.apellidos, (AVG(p."importe total")) AS "valor medio"

FROM Clientes c, Pedidos p

WHERE c.DNI = p.cliente

GROUP BY c.DNI, c.nombre, c.apellidos

**● Salida:**

**

*8. Muestra, para cada restaurante (código y nombre), el número total de platos vendidos y el precio acumulado que obtuvieron.*

**● Consulta:**

SELECT c.restaurante, r.nombre, (COUNT (DISTINCT c.plato)) AS numPlatos, (SUM(pl.precio)) AS precioAcum

FROM Restaurantes r, Platos pl, Pedidos p, Contiene c

WHERE r.código = pl.restaurante

AND r.código = c.restaurante

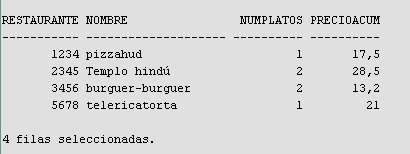
AND pl.nombre = c.plato

AND p.código = c.pedido

AND (p.estado = 'ENTREGADO' OR p.estado = 'REST')

GROUP BY c.restaurante, r.nombre

**● Salida:**

**

*9. Nombre y apellidos de aquellos clientes que pidieron platos de más de 15 €.*

**● Consulta:**

SELECT DISTINCT C.nombre, C.apellidos

FROM Clientes c, Pedidos p, Contiene co, Platos pl

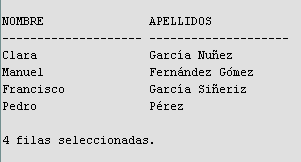
WHERE c.DNI = p.cliente

AND p.código = co.pedido

AND co.plato = pl.nombre

AND pl.precio > 15

**● Salida:**

**

*10. Para cada cliente, contar cuántos restaurantes cubren el área en el que el cliente vive.*

**● Consulta:**

SELECT c.DNI, (COUNT (DISTINCT restaurante)) AS numRestaurantes

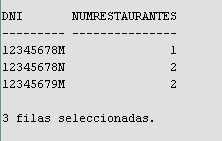
FROM Clientes c, "Areas Cobertura" a, Restaurantes r

WHERE c."código postal" = a."código postal"

AND a.restaurante = r.código

GROUP BY c.DNI

**● Salida:**

**

**NOTA:** Téngase en cuenta que algunas de las capturas de los resultados contienen información “no esperada” resultado de los INSERT INTO realizados a modo de prueba en el apartado 1 de creación de tablas. En concreto en los puntos 1,2,6,7,9