**AEV 3**

Muestra todos los campos de los clientes que no tienen teléfono asignado.

SELECT \*

FROM clientes

WHERE Teléfono = 0

Muestra por el DNI de cliente cuántas mascotas tiene cada cliente, el campo del conteo debe llamarse Cantidad. La consulta tiene que estar ordenada de mayor a menor cantidad de mascotas.

SELECT Dni\_Cliente, COUNT(Codigo) AS Cantidad

FROM mascotas

GROUP BY Dni\_Cliente ORDER BY Cantidad DESC

Muestra todos los campos de los clientes de Valencia que no tienen mascotas.

SELECT \*

FROM clientes

WHERE Población = 'Valencia' AND Dni NOT IN (

SELECT Dni\_Cliente

FROM mascotas)

Muestra todos los campos de las mascotas que han sido vacunadas después del 01/01/2021.

SELECT \*

FROM mascotas

WHERE Codigo IN (

SELECT Cod\_Mascota

FROM consultas

WHERE Fecha > 2021-01-01)

Muestra la cantidad de clientes que hay por población y el máximo de edad por población, pero solo los que sean mayores de 40 años.

SELECT Población, COUNT(Dni), MAX(Edad)

FROM clientes

WHERE Edad > 40

GROUP BY Población

Muestra todos los campos de los tratamientos que son más caros que la media de los tratamientos (precio).

SELECT \*

FROM tratamientos

WHERE Precio > (

SELECT AVG(Precio)

FROM tratamientos)

Muestra todos los campos de los tratamientos que no han sido contratados por nadie.

SELECT \*

FROM tratamientos

WHERE Cod\_Tratamiento NOT IN (

SELECT Cod\_Tratamiento

FROM consultas)

Muestra el código, nombre, raza y cuánto dinero se ha gastado cada mascota en sus consultas.

SELECT Codigo, Nombre, Raza, SUM(Precio)

FROM mascotas m

INNER JOIN consultas c

INNER JOIN tratamientos t ON m.codigo = c.Cod\_Mascota

AND c.Cod\_Tratamiento = t.Cod\_Tratamiento

GROUP BY Codigo

Muestra el nombre de las mascotas que contengan una hache en su nombre y que hayan realizado alguna consulta.

SELECT Nombre

FROM mascotas

WHERE Codigo = ANY (

SELECT Cod\_Mascota

FROM consultas)

AND nombre LIKE'%h%'

Muestra el nombre del cliente y el nombre de la mascota, de las mascotas que se han realizado una placa o una ecografía.

SELECT c.Nombre, m.Nombre

FROM clientes c

INNER JOIN mascotas m

INNER JOIN consultas s ON c.Dni = m.Dni\_Cliente

AND m.Codigo = s.Cod\_Mascota

WHERE Cod\_Tratamiento = 8 OR Cod\_Tratamiento = 9

Muestra el nombre de las mascotas que hayan gastado más de la media del precio en las consultas.

SELECT Nombre

FROM mascotas

WHERE Codigo = ANY (

SELECT Cod\_Mascota

FROM consultas

WHERE Cod\_Tratamiento = ANY (

SELECT Cod\_Tratamiento

FROM tratamientos

WHERE Precio > (

SELECT AVG(Precio)

FROM tratamientos)))

Muestra todos los campos de los clientes que tienen menos de 40 años y tienen perros.

SELECT \*

FROM clientes

WHERE Edad < 40 AND Dni IN (

SELECT Dni\_cliente

FROM mascotas

WHERE Raza = 'perro')

Muestra el nombre de las mascotas (sin que salga repetido el nombre) que han ido a consulta en enero de 2021.

SELECT m.Nombre, c.Fecha

FROM mascotas m

INNER JOIN consultas c

ON m.Codigo = c.Cod\_Mascota

WHERE fecha>='2021-01-01' AND fecha <='2021-01-31'

Muestra el nombre de mascota y la raza, ordenado por el nombre, de las que hayan realizado más de una consulta.

SELECT m.Nombre, m.Raza, c.ID\_Consulta

FROM mascotas m

INNER JOIN consultas c ON m.Codigo = c.Cod\_Mascota

GROUP BY m.Codigo, m.Nombre, m.Raza

HAVING COUNT(c.ID\_Consulta) > 1

ORDER BY m.Nombre

Muestra el nombre y la raza de las mascotas que sus clientes son de Alaquàs.

SELECT m.Nombre, m.Raza, c.Población

FROM mascotas m

INNER JOIN clientes c

ON m.Dni\_Cliente = c.Dni

WHERE c.Población = 'Alaquas'

**AEV 4**

Crea la siguiente base de datos por comandos.

Diagrama

El contenido generado por IA puede ser incorrecto.

CREATE DATABASE universidad

Crea las tablas necesarias para la base de datos, con las siguientes características y la eliminación de la relación será de tipo cascada

|  |  |
| --- | --- |
| **TABLA CURSO** | |
| **Campo** | **Tipos datos** |
| **Código** | Int (4) / Auto Incrementado / Primary Key |
| **Tipo** | Varchar (15) |
| **Nombre** | Varchar (30) |
| **Horas** | Int (3) |

|  |  |
| --- | --- |
| **TABLA PROFESORES** | |
| **Campo** | **Tipos datos** |
| **DNI** | Varchar(9) / Primary Key |
| **Nombre** | Varchar (15) |
| **Apellidos** | Varchar (20) |
| **Población** | Varchar (20), default ‘Valencia’ |
| **Teléfono** | Varchar (9) |

CREATE TABLE Profesores (

Dni VARCHAR(9) PRIMARY KEY,

Nombre VARCHAR(15),

Apellidos VARCHAR(20),

Poblacion VARCHAR(20) DEFAULT 'Valencia',

Telefono VARCHAR(9))

CREATE TABLE Curso (

Codigo INT(4) AUTO\_INCREMENT PRIMARY KEY,

Tipo VARCHAR(15),

Nombre VARCHAR(30),

Horas INT(3),

Dni\_profesor VARCHAR(9),

CONSTRAINT fk\_Dniprofesor

FOREIGN KEY (Dni\_profesor)

REFERENCES Profesores(Dni)

ON DELETE CASCADE )

Actualiza la tabla de PROFESORES añadiendo el campo EDAD como entero de 2 dígitos.

ALTER TABLE Profesores ADD COLUMN Edad INT(2)

Elimina el campo TELÉFONO de la tabla PROFESORES.

ALTER TABLE Profesores DROP COLUMN Telefono

Añade la tabla Alumnos con las siguientes características.

|  |  |  |
| --- | --- | --- |
| **TABLA ALUMNOS** | | |
| **Campo** | **Tipos datos** |
| **DNI** | Varchar(9) / Primary Key |
| **Nombre** | Varchar (15) |
| **Apellidos** | Int (2) |
| **Población** | Varchar (20) |
| **Teléfono** | Varchar (9) |

CREATE TABLE Alumnos (

Dni VARCHAR(9) PRIMARY KEY,

Nombre VARCHAR(15),

Apellidos INT(2),

Poblacion VARCHAR(20),

Telefono VARCHAR(9))

|  |  |
| --- | --- |
| **TABLA MATRÍCULA** | |
| **Campo** | **Tipos datos** |
| **Cod\_matric** | Int (4) / Auto Incrementado / Primary Key |
| **DNI\_alum** | Varchar (9) Relación con la tabla alumnos (eliminación tipo cascada) |
| **Cod\_curso** | Int(4) Relación con la tabla cursos (eliminación tipo cascada) |

Añade la tabla siguiente, el nombre será de MATRÍCULA:

CREATE TABLE Matricula (

Cod\_matric INT(4) AUTO\_INCREMENT PRIMARY KEY,

Dni\_alum VARCHAR(9),

CONSTRAINT fk\_dnialumno

FOREIGN KEY (Dni\_alum)

REFERENCES alumnos(Dni)

ON DELETE CASCADE,

Cod\_curso INT(4),

CONSTRAINT fk\_codigocurso

FOREIGN KEY (Cod\_curso)

REFERENCES curso(Codigo)

ON DELETE CASCADE)

Modifica en la tabla Alumnos el tipo del campo Apellidos a Varchar (20).

ALTER TABLE alumnos MODIFY COLUMN Apellidos VARCHAR(20)

Modifica el campo de Población de la tabla Alumnos, para que salga por defecto ‘Valencia’.

ALTER TABLE alumnos ALTER COLUMN Poblacion SET DEFAULT 'Valencia'

Modifica la tabla alumnos añadiendo el campo de Fecha\_Nacimiento que sea del tipo date.

ALTER TABLE alumnos ADD COLUMN Fecha\_nacimiento DATE

Rellena las tablas con los siguientes datos:

* TABLA ALUMNOS:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DNI** | **NOMBRE** | **APELLIDOS** | **POBLACIÓN** | **TELÉFONO** | **FEC\_NAC** |
| 60655544P | MARÍA | FERRER GARCÍA | VALENCIA | 666555444 | 15/02/2000 |
| 00655544P | MARÍA | FERRER GARCÍA | VALENCIA | 606555444 | 11/10/2003 |
| 10655544P | JOSE | MARTÍNEZ GARCÍA | ALAQUAS | 616555444 | 15/06/1990 |
| 20655544P | MARÍA | GONZÁLEZ GARCÍA | TORRENT | 626555444 | 09/09/1996 |
| 30655544P | MARC | SAEZ GONZÁLEZ | TORRENT | 636555444 | 08/02/1998 |
| 40655544P | MARTA | FERRER MARTÍNEZ | VALENCIA | 646555444 | 04/04/1993 |
| 50655544P | FRANCISCO | CASAS GARCÍA | PATERNA | 656555444 | 21/03/1980 |
| 60655544Q | MARCOS | FERRER SAEZ | VALENCIA | 676555444 | 22/06/1967 |
| 70655544P | RAUL | GONZÁLEZ GARCÍA | VALENCIA | 686555444 | 25/07/1964 |
| 80655544P | JOSE | CASAS MARTÍN | PATERNA | 696555444 | 15/02/1960 |
| 90655544P | MARIO | FERRER GONZÁLEZ | ALAQUAS | 666555440 | 02/05/1985 |
| 60655544R | CATALINA | MARTÍN COLON | PATERNA | 666555441 | 12/02/2000 |
| 60655544D | MARÍA | CID GARCÍA | TORRENT | 666555442 | 10/11/2001 |
| 60655544V | MARCOS | FERRER SANZ | VALENCIA | 676555443 | 15/02/2001 |
| 60655544F | RODRIGO | GONZÁLEZ CAMARA | XIRIVELLA | 686555445 | 15/12/1990 |
| 60655544X | JOSE | CASAS MARTÍN | PATERNA | 696555446 | 15/02/1998 |
| 60655544A | MARTÍN | CAMARA GONZÁLEZ | XIRIVELLA | 666555447 | 15/02/2003 |
| 60655544U | CARMEN | SANZ CASAS | PATERNA | 666555448 | 13/02/2001 |
| 60655544N | RODRIGO | FERRER CID | TORRENT | 666555449 | 15/03/1968 |

INSERT INTO Alumnos VALUES

('60655544P', 'MARÍA', 'FERRER GARCÍA', 'VALENCIA', '666555444', '2000-02-15'),

('00655544P', 'MARÍA', 'FERRER GARCÍA', 'VALENCIA', '606555444', '2003-10-11'),

* TABLA PROFESORES:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DNI** | **NOMBRE** | **APELLIDOS** | **POBLACIÓN** | **EDAD** |
| 10655544P | RAQUEL | FERNÁNDEZ REVUELTA | VALENCIA | 46 |
| 11655544P | NURIA | MARTÍNEZ SAEZ | ALDAIA | 50 |
| 12655544P | JOSE | CUENCA MÁS | VALENCIA | 56 |
| 13655544P | FRANCISCO | ROJO SAEZ | TORRENT | 43 |
| 14655544P | CARLOS | TORO CASAS | VALENCIA | 29 |
| 15655544P | DAVID | LEÓN RODRÍGUEZ | VALENCIA | 35 |
| 16655544P | DAVID | GARCÍA FERRER | ALDAIA | 38 |
| 17655544P | FERNANDO | FERRER CÁMARA | TORRENT | 42 |
| 18655544P | MARTA | AMARILLO RUIZ | PAIPORTA | 59 |
| 19655544P | MIGUEL | SANZ LÓPEZ | TORRENT | 60 |

INSERT INTO Profesores VALUES

('10655544P', 'RAQUEL', 'FERNÁNDEZ REVUELTA', 'VALENCIA', '46'),

('11655544P', 'NURIA', 'MARTÍNEZ SAEZ', 'ALDAIA', '50'),

* TABLA CURSOS:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CODIGO** | **TIPO** | **NOMBRE** | **HORAS** | **DNI\_PROF** |
| 1 | INFORMÁTICA | PROGRAMACIÓN C++ | 500 | 11655544P |
| 2 | INFORMÁTICA | PROGRAMACIÓN PHP | 600 | 12655544P |
| 3 | INFORMÁTICA | PROGRAMACIÓN JAVA | 400 | 11655544P |
| 4 | INFORMÁTICA | APLICACIONES OFIMÁTICAS | 350 | 12655544P |
| 5 | MARKETING | COMERCIO INTERNACIONAL | 200 | 13655544P |
| 6 | MARKETING | ECONOMÍA | 300 | 15655544P |
| 7 | MARKETING | RRHH | 500 | 15655544P |
| 8 | MECÁNICA | FORMULA 1 | 550 | 11655544P |
| 9 | MECÁNICA | CIRCUITOS | 700 | 14655544P |
| 10 | LOGÍSTICA | TRANSPORTES | 600 | 17655544P |
| 11 | LOGÍSTICA | COMERCIO INTERNACIONAL | 100 | 16655544P |

INSERT INTO Curso (Tipo, Nombre, Horas, Dni\_profesor) VALUES

('INFORMÁTICA', 'PROGRAMACIÓN C++', 500, '11655544P'),

('INFORMÁTICA', 'PROGRAMACIÓN PHP', 600, '12655544P'),

('INFORMÁTICA', 'PROGRAMACIÓN JAVA', 400, '11655544P'),

* TABLA MATRÍCULA:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **DNI\_ALUM** | **COD\_CURSO** |  | **CODIGO** | **DNI\_ALUM** | **COD\_CURSO** |
| NULL | 40655544P | 1 |  | NULL | 00655544P | 2 |
| NULL | 50655544P | 11 |  | NULL | 50655544P | 11 |
| NULL | 60655544P | 10 |  | NULL | 60655544F | 10 |
| NULL | 60655544P | 1 |  | NULL | 60655544D | 2 |
| NULL | 60655544P | 3 |  | NULL | 60655544U | 9 |
| NULL | 10655544P | 2 |  | NULL | 60655544Q | 2 |
| NULL | 60655544P | 1 |  | NULL | 60655544X | 6 |
| NULL | 60655544P | 10 |  | NULL | 00655544P | 6 |
| NULL | 60655544P | 8 |  | NULL | 60655544X | 5 |
| NULL | 60655544P | 8 |  | NULL | 20655544P | 3 |
| NULL | 10655544P | 8 |  | NULL | 30655544P | 3 |
| NULL | 90655544P | 7 |  | NULL | 70655544P | 2 |
| NULL | 80655544P | 6 |  | NULL | 90655544P | 1 |

INSERT INTO Matricula (Dni\_alum, Cod\_curso) VALUES

('40655544P', 1), ('00655544P', 2), ('50655544P', 11), ('50655544P', 11), ('80655544P', 6);

Actualiza la tabla de PROFESORES, todos los que tengan menos de 40 años y son de Valencia pasan a ser de Catarroja.

UPDATE profesores SET Poblacion = 'Catarroja' WHERE Edad < 40 AND Poblacion = 'Valencia'

Actualiza la tabla de CURSOS, todos los cursos que tengan más de 300 horas van a tener un 10% más de horas.

UPDATE curso SET Horas = Horas \* 1.10 WHERE Horas > 300

Crea una tabla con los alumnos que nacieron antes del 2000, se llamará alumnos2000.

CREATE TABLE alumnos2000

SELECT \*

FROM alumnos

WHERE Fecha\_nacimiento < '2000-01-01'

Elimina de la tabla alumnos2000, los alumnos que han nacido después de 1990 y son de Torrent.

DELETE FROM alumnos2000

WHERE YEAR(Fecha\_nacimiento) > 1990 AND Poblacion = 'Torrent'

Crea una tabla con los alumnos que hayan nacido después del año 2000, incluido el 2000. Se llamará Juveniles.

CREATE TABLE juveniles

SELECT \*

FROM alumnos

WHERE YEAR(Fecha\_nacimiento) >= '2000'

Anexa a la tabla juveniles de la tabla alumnos2000, pero solo los que hayan nacido en Valencia.

INSERT INTO juveniles

SELECT \*

FROM alumnos2000

WHERE Poblacion = 'Valencia'

Actualiza la tabla Matrícula, todas las matrículas que se han del tipo informática hay que sumarle uno al código curso.

UPDATE matricula SET Cod\_curso = Cod\_curso + 1 WHERE Cod\_curso IN (SELECT codigo FROM curso WHERE Tipo = 'Informática')

Eliminar todos los alumnos que estén matriculados en marketing.

DELETE FROM alumnos

WHERE Dni IN (

SELECT Dni\_alum

FROM matricula

WHERE Cod\_curso IN (

SELECT Cod\_curso

FROM curso

WHERE Tipo = 'Marketing'

)

)