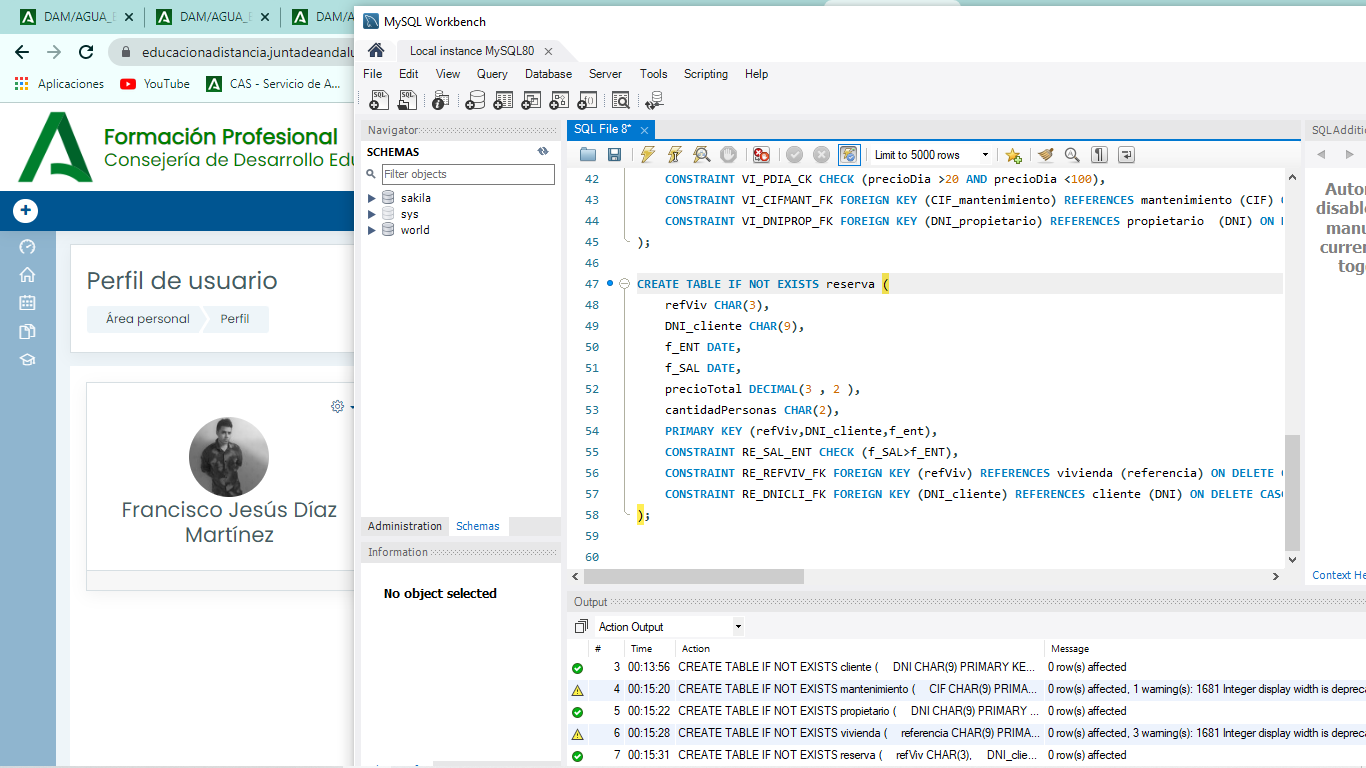
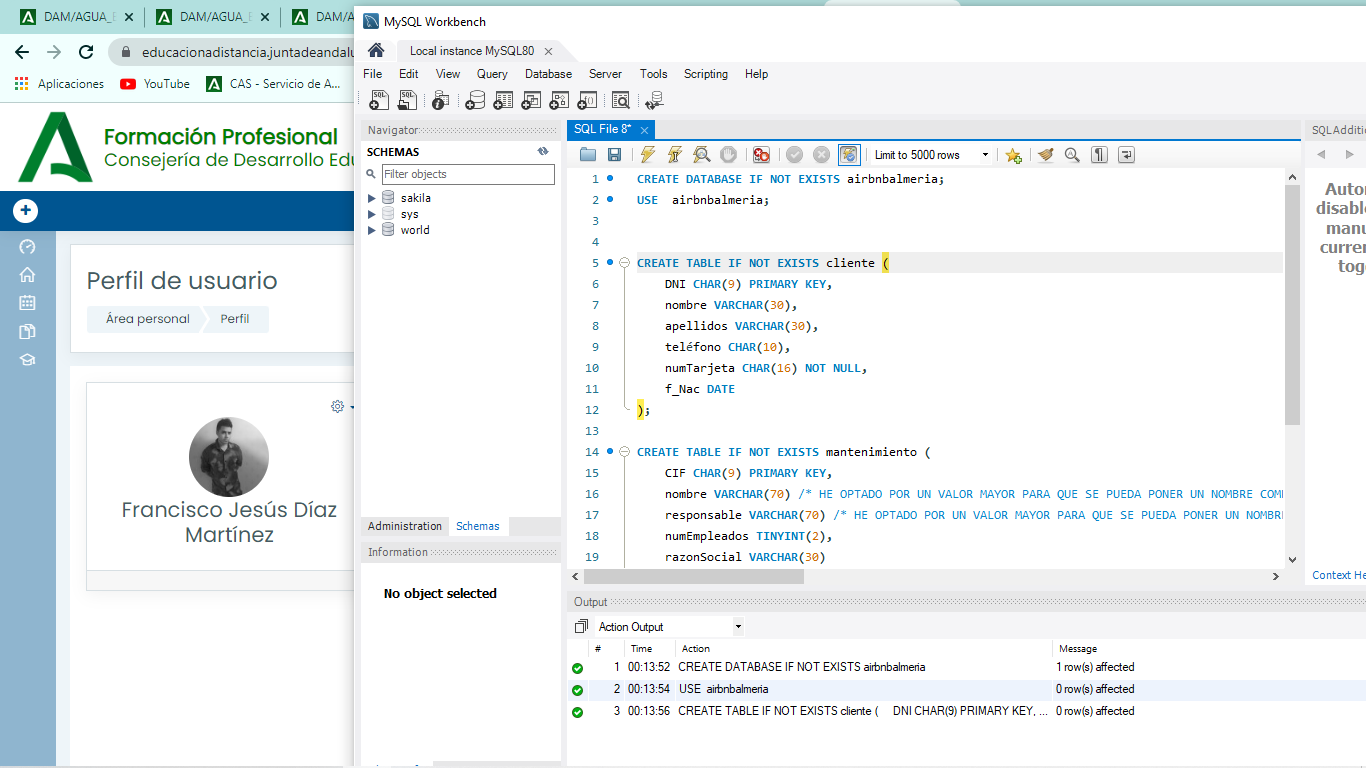
bases de datos

CREACIÓN DE TABLAS SQL (UNIDAD 3)

IES AGUADULCE, DAM 1A

FRANCISCO JESÚS DÍAZ MARTÍNEZ



**APARTADO A:**

Aclaración; Los comentarios tipo /\*--\*/ de color azul solo están disponibles en este documento de texto a modo explicativo. Los de color gris estan incluidos en el SQL.

/\* En primer lugar creamos la base de datos, y la inicializamos \*/

**CREATE** **DATABASE** IF **NOT** **EXISTS** airbnbalmeria;

**USE** airbnbalmeria;

/\* Para ser consecuentes, creamos en primer lugar las tablas que no dependen de otras. Tienen sus claves primarias, y son referencia para las claves ajenas de las tablas “vivienda” y “reserva” \*/

/\* En la tabla “cliente”, campo “numTarjeta”, se ha especificado el valor NOT NULL porque en el enunciado de la tarea nos dicen que no puede quedarse vacío \*/

**CREATE** **TABLE** IF **NOT** **EXISTS** cliente (

DNI CHAR(**9**) **PRIMARY** **KEY**,

nombre VARCHAR(**30**),

apellidos VARCHAR(**30**),

teléfono CHAR (**10**),

numTarjeta CHAR(**16**) **NOT** **NULL**,

f\_Nac DATE

);

**CREATE** **TABLE** IF **NOT** **EXISTS** mantenimiento (

CIF CHAR(**9**) **PRIMARY** **KEY**,

nombre VARCHAR(**70**) */\* HE OPTADO POR UN VALOR MAYOR PARA QUE SE PUEDA PONER UN NOMBRE COMPLETO EN EL MISMO CAMPO\*/*,

responsable VARCHAR(**70**) */\* HE OPTADO POR UN VALOR MAYOR PARA QUE SE PUEDA PONER UN NOMBRE COMPLETO EN EL MISMO CAMPO\*/*,

numEmpleados TINYINT(**2**),

razonSocial VARCHAR(**30**)

);

/\* En la tabla “propietario”, campo “beneficio”, hemos optado por darle un valor decimal, suponiendo que puede tener un beneficio de 4 cifras y dos decimales, también entendemos que si ha arrendado su casa tiene beneficio y este campo no puede quedarse vacío, por ello hemos indicado NOT NULL. También le hemos dado la cualidad UNIQUE al numCuenta porque nos han especificado que el valor que contenga este campo es único y no se puede repetir, así por último le hemos dado la longitud correcta de caracteres (24) porque va a contener un IBAN \*/

**CREATE** **TABLE** IF **NOT** **EXISTS** propietario (

DNI CHAR(**9**) **PRIMARY** **KEY**,

nombre VARCHAR(**30**),

apellidos VARCHAR(**40**),

numCuenta CHAR(**24**) **UNIQUE** */\*SE HA ESPECIFICADO 24 CARACTERES PORQUE SON LOS QUE COMPONEN EL IBAN\*/*,

beneficio DECIMAL(**4** , **2** ) **NOT** **NULL**

);

/\* En la tabla vivienda, la más larga hasta ahora podemos observar varias restricciones y claves ajenas. Para empezar, tenemos un CHECK para el campo precioDia que comprueba que el valor que introduzcamos sea mayor de 20 y menor de 100. También dos claves ajenas CIF\_mantenimiento y DNI\_propietario, que apuntan a las tablas principales “mantenimiento” y “propietario” respectivamente. A las tres restricciones se le ha dado un valor personalizado mediante el comando “CONSTRAINT” para que puedan ubicarse fácilmente en caso de necesitarse. Por último tenemos tres valores n\_habitaciones,n\_camas y n\_aseos a los que se la ha dado un valor (1) por defecto mediante el comando DEFAULT \*/

**CREATE** **TABLE** IF **NOT** **EXISTS** vivienda (

referencia CHAR(**9**) **PRIMARY** **KEY**,

direccion\_vivienda VARCHAR(**40**),

localidad VARCHAR(**30**),

provincia VARCHAR(**30**),

n\_habitaciones TINYINT(**1**) **DEFAULT** **1**,

n\_camas TINYINT(**1**) **DEFAULT** **1**,

n\_aseos TINYINT(**1**) **DEFAULT** **1**,

metros2 VARCHAR(**3**),

precioDia DECIMAL(**3** , **2** ) */\*SE HA ESPECIFICADO UN TIPO DECIMAL SUPONIENDO QUE LA CIFRA SERÁ DEL FORMATO XXX,XX €\*/*,

CIF\_mantenimiento CHAR(**9**),

DNI\_propietario CHAR(**9**),

**CONSTRAINT** VI\_PDIA\_CK **CHECK** (precioDia >**20** **AND** precioDia <**100**),

**CONSTRAINT** VI\_CIFMANT\_FK **FOREIGN** **KEY** (CIF\_mantenimiento) **REFERENCES** mantenimiento (CIF) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**CONSTRAINT** VI\_DNIPROP\_FK **FOREIGN** **KEY** (DNI\_propietario) **REFERENCES** propietario (DNI) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

/\* En esta última tabla, encontramos también 2 claves ajenas y una clave primaria compuesta por 3 campos. Al igual que en la tabla anterior llevan asociada la instrucción “ON DELETE CASCADE ON UPDATE CASCADE” que tiene la función de interactuar con el campo original al que apuntan, actualizando u eliminando el valor según se haga en la tabla de referencia. También hay un “CHECK” que comprueba que la fecha de salida sea siempre posterior a la de entrada \*/

**CREATE** **TABLE** IF **NOT** **EXISTS** reserva (

refViv CHAR(**3**),

DNI\_cliente CHAR(**9**),

f\_ENT DATE,

f\_SAL DATE,

precioTotal DECIMAL(**3** , **2** ),

cantidadPersonas CHAR(**2**)/\* Se han especificado un valor de 2 en caso de ser más de 9 personas para casas tipo chalet \*/,

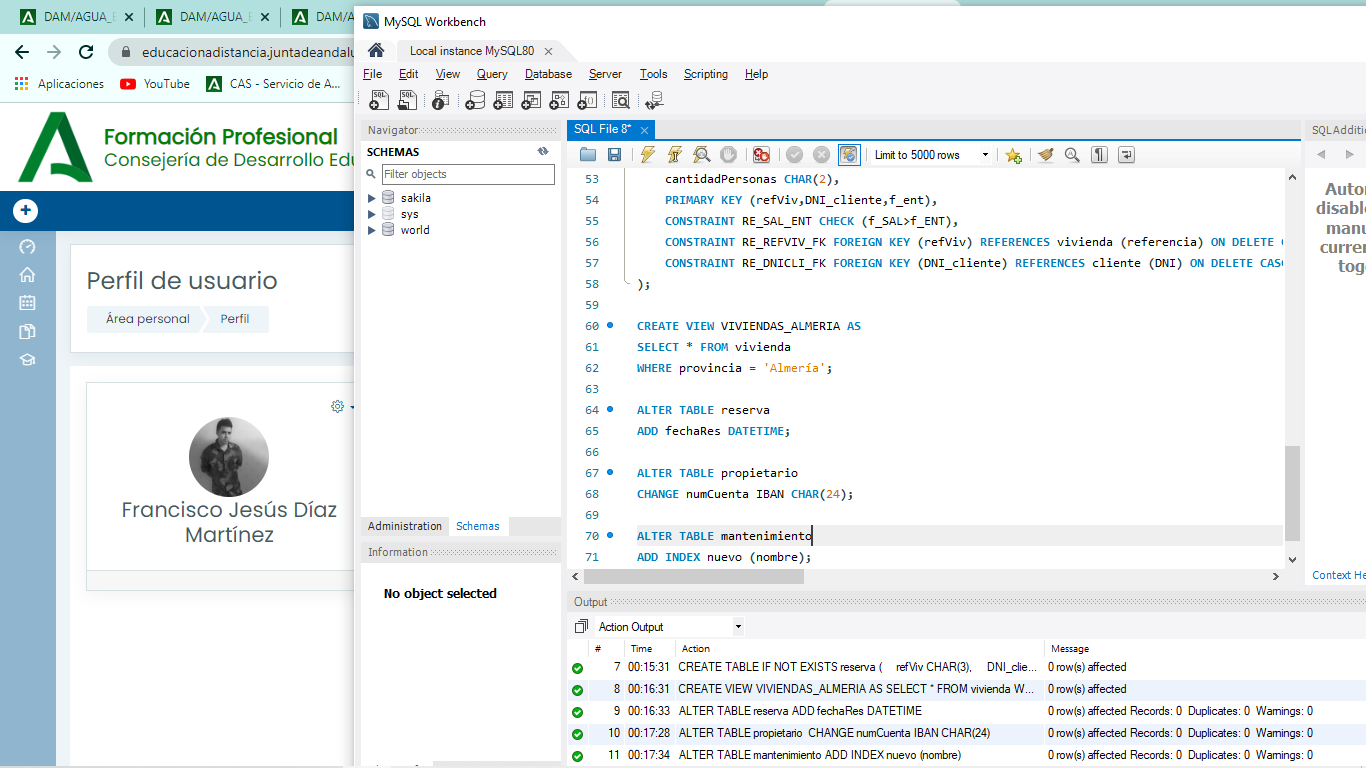
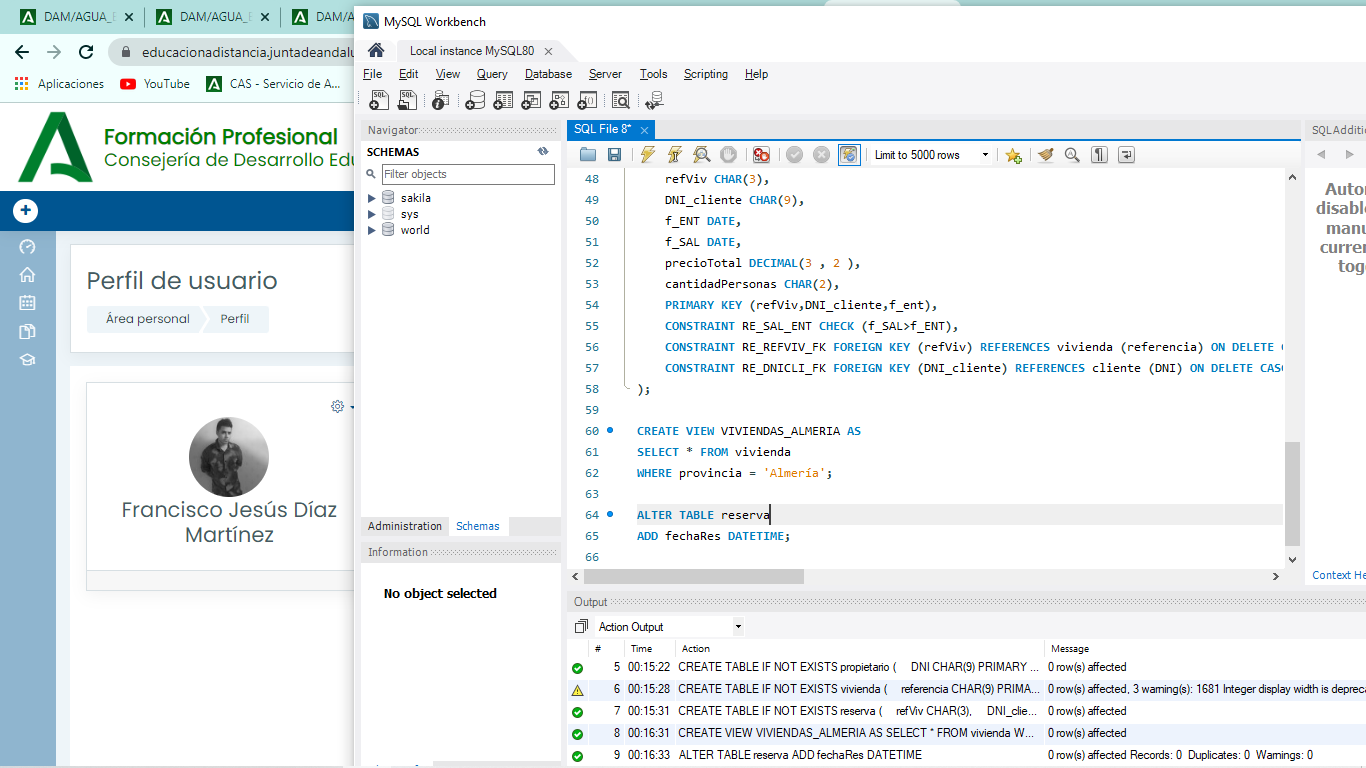
**PRIMARY** **KEY** (refViv,DNI\_cliente,f\_ent),

**CONSTRAINT** RE\_SAL\_ENT **CHECK** (f\_SAL>f\_ENT),

**CONSTRAINT** RE\_REFVIV\_FK **FOREIGN** **KEY** (refViv) **REFERENCES** vivienda (referencia) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**CONSTRAINT** RE\_DNICLI\_FK **FOREIGN** **KEY** (DNI\_cliente) **REFERENCES** cliente (DNI) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);



**APARTADO B**

En este apartado el enunciado nos solicita que modifiquemos las tablas creadas anteriormente, creando una vista, añadiendo un campo, cambiando el nombre de un campo y por último creando un índice.

Explicación de las tablas y parámetros elegidos en la siguiente hoja.

/\* Comenzamos por la creación de la vista. Mediante el comando CREATE VIEW, con la identificación VIVIENDAS\_ALMERIA, indicamos que seleccione todos los valores que coincidan con ‘Almeria’ para la tabla vivienda y concretamente en el campo ‘provincia’ \*/

**CREATE** **VIEW** VIVIENDAS\_ALMERIA **AS**

**SELECT** \* **FROM** vivienda

**WHERE** provincia = 'Almería';

/\* En esta ocasión nos indica el enunciado que agreguemos el campo “fechaRes” a la tabla reserva. Le añadimos el tipo de dato DATETIME que indica un formato AAAA-MM-DD \*/

**ALTER** **TABLE** reserva

**ADD** fechaRes **DATETIME**;

/\* Para la table propietario necesitamos cambiar el campo definido como “numCuenta” por el nuevo nombre “IBAN”. Indicamos de nuevo que es un valor tipo CHAR de 24 dígitos pues este es el formato de las cuentas bancarias \*/

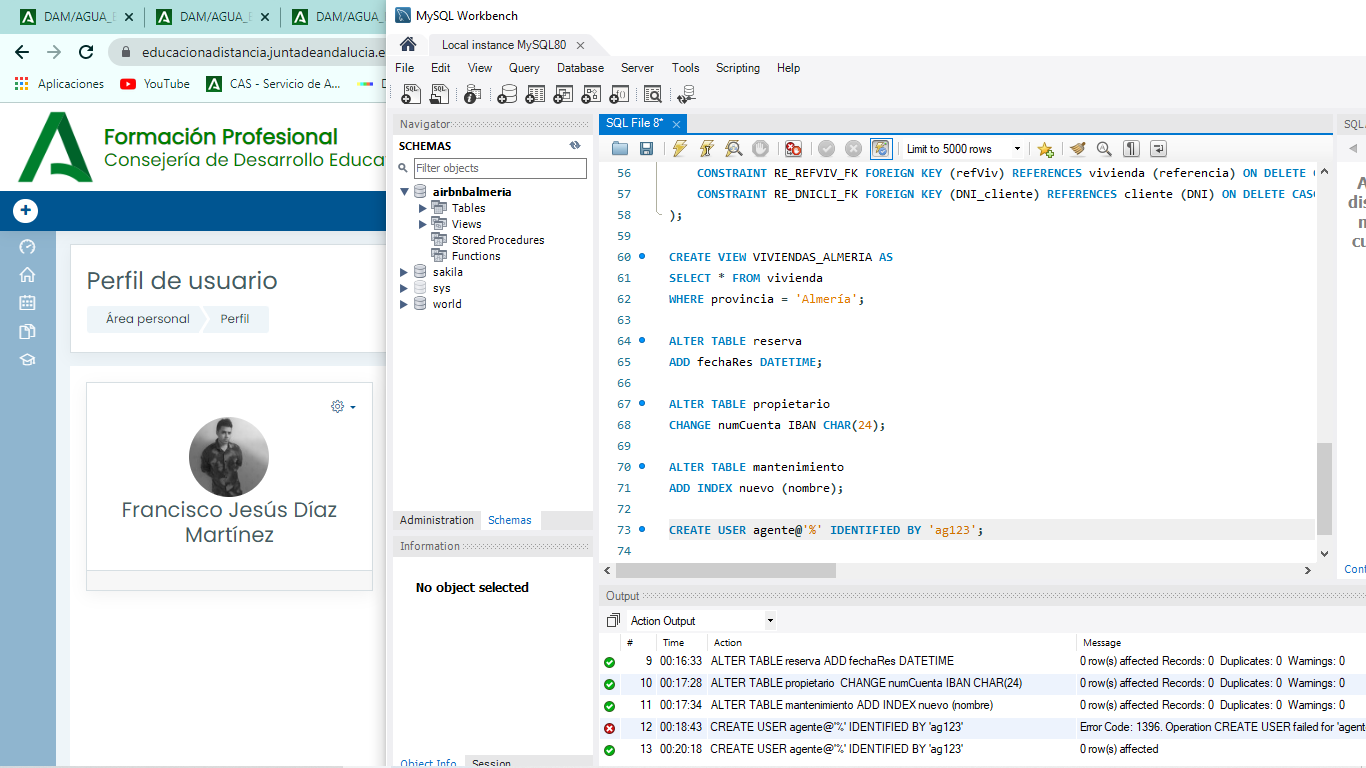
**ALTER** **TABLE** propietario

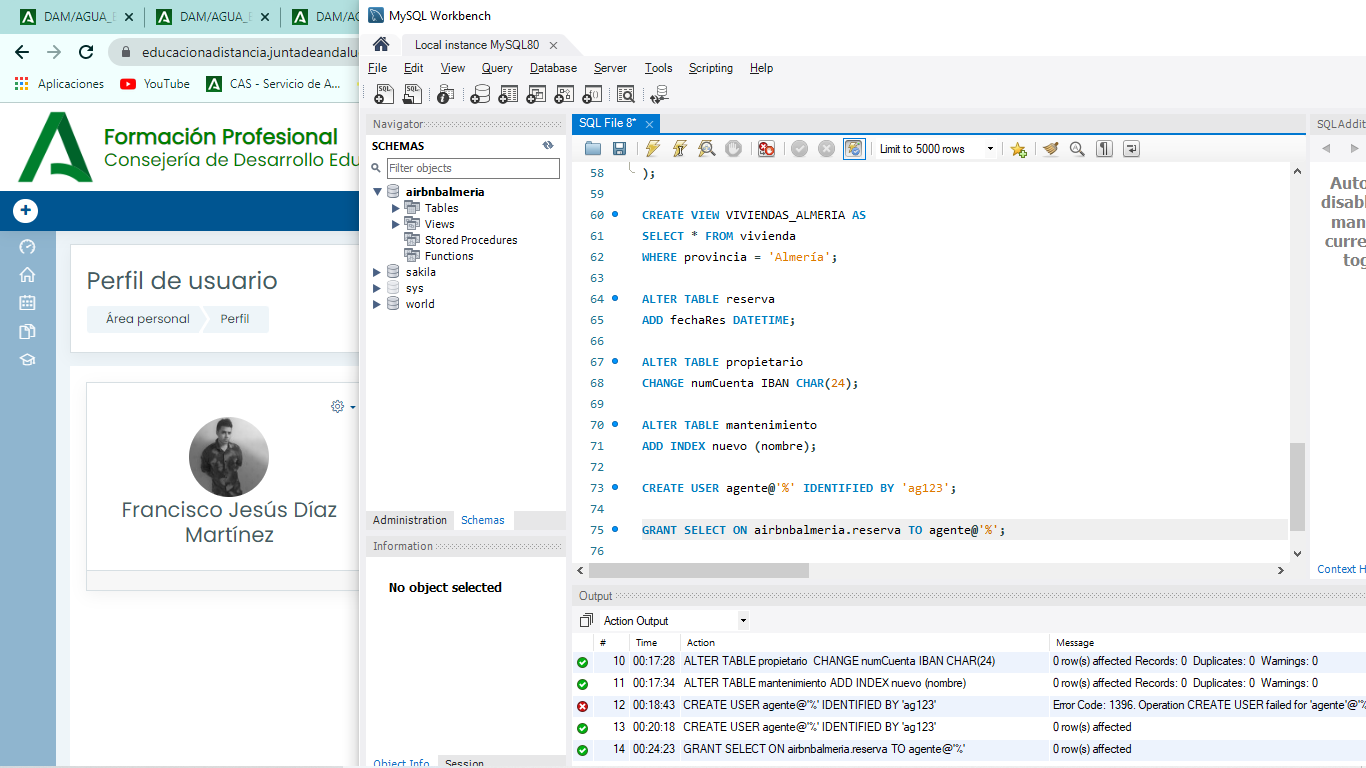
**CHANGE** numCuenta IBAN CHAR(**24**);

/\* Nos solicitan la creación de un índice, en la tabla “mantenimiento” para el campo “nombre”. Para ello seleccionamos la tabla a la que hacemos referencia con el comando ALTER TABLE, y con ADD INDEX apuntaos a la tabla y campo que sea preciso \*/

**ALTER** **TABLE** mantenimiento

**ADD** **INDEX** nuevo (nombre);





**APARTADO C**

Para finalizar, se nos solicita que creemos un usuario para esta base de datos y que le demos ciertos permisos, veamos cuales son las sentencias para ello.

/\* Nos dan el usuario y la contraseña con la que tenemos que crear el usuario. Nos dicen que debe de poder acceder desde cualquier equipo. Con CREATE USER abrimos la sentencia para dar nombre al usuario, seguido de @ y ‘%’ que implica que puede acceder desde cualquier sitio. Posteriormente establecemos la contraseña que nos han indicado escribiéndola después de IDENTIDIED BY \*/

**CREATE** **USER** agente@'%' **IDENTIFIED** **BY** 'ag123';

/\*Finalmente, el usuario necesita poder acceder y visualizar los contenidos de la tabla reserva. La sentencia para permitir esto comienza con GRANT SELECT ON, seguido del nombre de la base datos (.) y el nombre de la tabla a la que necesita tener acceso. A continuación mediante el TO se indica a que usuario en cuestión se le otorga este acceso\*/

**GRANT** **SELECT** **ON** airbnbalmeria.reserva **TO** agente@'%';