

Base de Datos

Práctica 1.

Objetivo:

Introducción al Mysql, aprender a identificar los errores más comunes en la creación de las bases de datos para su solución, conocer los diferentes tipos de datos y la importancia de utilizar el tipo de dato adecuado y aprender a utilizar los comando para la creación de las bases de datos y tablas y la manipulación de las mismas:

- **Show databases**
- **Create database**
- **Create table**
- **Use table**
- **Show tables**
- **Describe table**
- **Show create table**
- **Drop table**
- **Alter table**
- **Insert into**

Para esta práctica se va a utilizar el manual de Mysql proporcionado, un servidor Linux con un SGBD de Mysql, y el programa Putty para acceder al servidor de Linux.

Metodología:

Crear, guiado por la profesora la base de datos constructora con las tablas, que se muestran a en la siguiente página, durante la creación se irán revisando cada uno de los comandos indicados en el objetivo, para los cuales se puede consultar el manual de Mysql o el internet..

Tables_in_CONSTRUCTORA
Asignacion
Empleado
Proyecto
Puesto

Proyecto			
<i>Field</i>	<i>Type</i>	<i>Null</i>	<i>Key</i>
Proy_ID	smallint(6)	NO	PRI
Proy_Name	varchar(30)	NO	

Asignacion			
<i>Field</i>	<i>Type</i>	<i>Null</i>	<i>Key</i>
Proy_ID	smallint(6)	NO	PRI
Emp_id	smallint(6)	NO	PRI
Asig_hrs	float(4,2)	NO	

Empleado			
<i>Field</i>	<i>Type</i>	<i>Null</i>	<i>Key</i>
Emp_ID	smallint(6)	NO	PRI
Emp_Name	varchar(30)	NO	
Emp_Apat	varchar(20)	NO	
Emp_Amat	varchar(20)	YES	
Pto_ID	smallint(6)	NO	MUL

Puesto			
<i>Field</i>	<i>Type</i>	<i>Null</i>	<i>Key</i>
Pto_ID	smallint(6)	NO	PRI
Pto_Empleo_tipo	varchar(30)	NO	
Pto_Pago_hr	float(7,2)	NO	

Base de Datos

Práctica 2.

Objetivo:

Comprender la integridad referencial y conocer el uso y razón de ser de las llaves primarias y foráneas, identificar las tablas padre e hijo en una relación, además de aprender a crear tablas implementando integridad referencial en MySQL.

Introducción:

Integridad referencial en MySQL

Autor: [MySQL Hispano](#)

- [Claves foráneas e integridad referencial](#)

http://www.programacion.com/bbdd/tutorial/mysql_referencial/3/

Claves foráneas e integridad referencial

Podemos decir de manera simple que integridad referencial significa que cuando un registro en una tabla haga referencia a un registro en otra tabla, el registro correspondiente debe existir. Por ejemplo, consideremos la relación entre una tabla cliente y una tabla venta.

```
+-----+ +-----+
| cliente | | venta   |
+-----+ +-----+
| id_cliente | | id_factura |
| nombre   | | id_cliente |
+-----+ | cantidad |
           +-----+
```

Para poder establecer una relación entre dos tablas, es necesario asignar un campo en común a las dos tablas. Para este ejemplo, el campo `id_cliente` existe tanto en la tabla cliente como en la tabla venta. La mayoría de las veces, este campo en común debe ser una clave primaria en alguna de las tablas. Vamos a insertar algunos datos en estas tablas.

Tabla cliente

```
+-----+
| id_cliente | nombre   |
```

1	Juan penas
2	Pepe el Toro

Tabla venta

id_factura	id_cliente	cantidad
1	1	23
2	3	39
3	2	81

Hay dos registros en la tabla cliente, pero existen 3 id_cliente distintos en la tabla venta. Habíamos dicho que las dos tablas se relacionan con el campo id_cliente, por lo tanto, podemos decir que Juan Penas tiene una cantidad de 23, y Pepe el Toro 81, sin embargo, no hay un nombre que se corresponda con el id_cliente 3.

Las relaciones de claves foráneas se describen como relaciones padre/hijo (en nuestro ejemplo, cliente es el padre y venta es el hijo), y se dice que un registro es huérfano cuando su padre ya no existe.

Cuando en una base de datos se da una situación como esta, se dice que se tiene una integridad referencial pobre (pueden existir otra clase de problemas de integridad). Generalmente esto va ligado a un mal diseño, y puede generar otro tipo de problemas en la base de datos, por lo tanto debemos evitar esta situación siempre que sea posible.

En el pasado, MySQL no se esforzaba en evitar este tipo de situaciones, y la responsabilidad pasaba a la aplicación. Para muchos desarrolladores, esta no era una situación del todo grata, y por lo tanto no se consideraba a MySQL para ser usado en sistemas "serios". Por supuesto, esta fue una de las cosas más solicitadas en las anteriores versiones de MySQL; que se tuviera soporte para claves foráneas, para que MySQL mantenga la integridad referencial de los datos.

Una clave foránea es simplemente un campo en una tabla que se corresponde con la clave primaria de otra tabla. Para este ejemplo, el campo id_cliente en la tabla venta es la clave foránea. Nótese que este campo se corresponde con el campo id_cliente en la tabla cliente, en dónde este campo es la clave primaria.

Las claves foráneas tienen que ver precisamente con la integridad referencial, lo que significa que si una clave foránea contiene un valor, ese valor se refiere a un registro existente en la tabla relacionada.

Integridad referencial en MySQL

Autor: [MySQL Hispano](#)

- [Claves foráneas en MySQL](#)

http://www.programacion.com/bbdd/tutorial/mysql_referencial/4/

Claves foráneas en MySQL

Estrictamente hablando, para que un campo sea una clave foránea, éste necesita ser definido como tal al momento de crear una tabla. Se pueden definir claves foráneas en cualquier tipo de tabla de MySQL, pero únicamente tienen sentido cuando se usan tablas del tipo InnoDB.

A partir de la versión 3.23.43b, se pueden definir restricciones de claves foráneas con el uso de tablas InnoDB. InnoDB es el primer tipo de tabla que permite definir estas restricciones para garantizar la integridad de los datos.

Para trabajar con claves foráneas, necesitamos hacer lo siguiente:

- Crear ambas tablas del tipo InnoDB.
- Usar la sintaxis FOREIGN KEY(campo_fk) REFERENCES nombre_tabla (nombre_campo)
- Crear un índice en el campo que ha sido declarado clave foránea.

InnoDB no crea de manera automática índices en las claves foráneas o en las claves referenciadas, así que debemos crearlos de manera explícita. Los índices son necesarios para que la verificación de las claves foráneas sea más rápida. A continuación se muestra como definir las dos tablas de ejemplo con una clave foránea.

```
CREATE TABLE cliente
(
    id_cliente INT NOT NULL,
    nombre VARCHAR(30),
    PRIMARY KEY (id_cliente)
) TYPE = INNODB;

CREATE TABLE venta
(
    id_factura INT NOT NULL,
    id_cliente INT NOT NULL,
    cantidad INT,
    PRIMARY KEY(id_factura),
    INDEX (id_cliente),
    FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente)
) TYPE = INNODB;
```

La sintaxis completa de una restricción de clave foránea es la siguiente:

```
[CONSTRAINT símbolo] FOREIGN KEY (nombre_columna, ...)
    REFERENCES nombre_tabla (nombre_columna, ...)
    [ON DELETE {CASCADE | SET NULL | NO ACTION
                | RESTRICT}]
    [ON UPDATE {CASCADE | SET NULL | NO ACTION
                | RESTRICT}]
```

Las columnas correspondientes en la clave foránea y en la clave referenciada deben tener tipos de datos similares para que puedan ser comparadas sin la necesidad de hacer una conversión de tipos. El tamaño y el signo de los tipos enteros debe ser el mismo. En las columnas de tipo carácter, el tamaño no tiene que ser el mismo necesariamente.

Si MySQL da un error cuyo número es el 1005 al momento de ejecutar una sentencia CREATE TABLE, y el mensaje de error se refiere al número 150, la creación de la tabla falló porque la restricción de la clave foránea no se hizo de la manera adecuada. De la misma manera, si falla una sentencia ALTER TABLE y se hace referencia al error número 150, esto significa que la definición de la restricción de la clave foránea no se hizo adecuadamente. A partir de la versión 4.0.13 de MySQL, se puede usar la sentencia SHOW INNODB STATUS para ver una explicación detallada del último error que se generó en relación a la definición de una clave foránea.

Si en una tabla, un registro contiene una clave foránea con un valor NULO, significa que no existe ninguna relación con otra tabla.

A partir de la versión 3.23.50, se pueden agregar restricciones de clave foránea a una tabla con el uso de la sentencia ALTER TABLE. La sintaxis es:

```
ALTER TABLE nombre_tabla ADD [CONSTRAINT símbolo] FOREIGN KEY(...)
REFERENCES otra_tabla(...) [acciones_ON_DELETE][acciones_ON_UPDATE]
```

Por ejemplo, la creación de la clave foránea en la tabla venta que se mostró anteriormente pudo haberse hecho de la siguiente manera con el uso de una sentencia ALTER TABLE:

```
CREATE TABLE venta
(
  id_factura INT NOT NULL,
  id_cliente INT NOT NULL,
  cantidad INT,
  PRIMARY KEY(id_factura),
  INDEX (id_cliente)
) TYPE = INNODB;

ALTER TABLE venta ADD FOREIGN KEY(id_cliente) REFERENCES cliente(id_cliente);
```

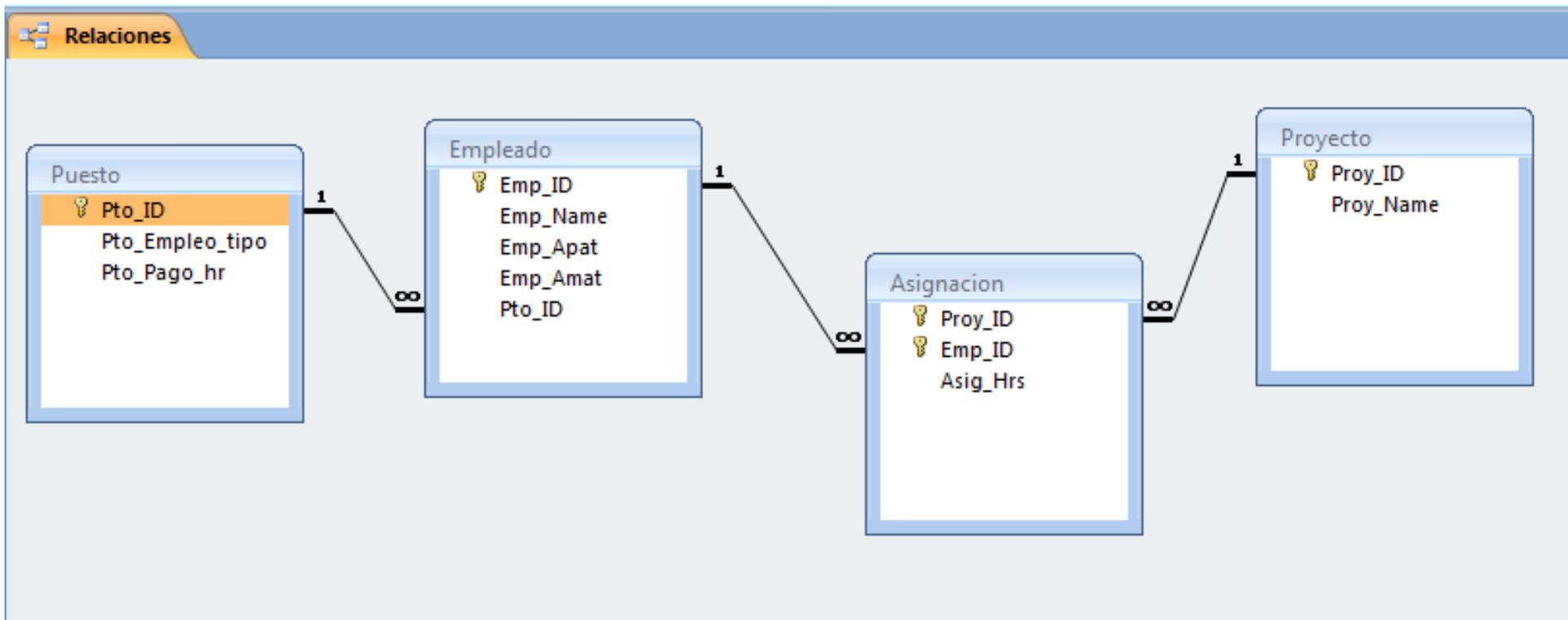
En las versiones 3.23.50 y menores no deben usarse las sentencias ALTER TABLE o CREATE INDEX en tablas que ya tienen definidas restricciones de claves foráneas o bien, que son referenciadas en restricciones de claves foráneas: cualquier sentencia ALTER TABLE elimina todas las restricciones de claves foráneas definidas para la tabla.

No debe usarse una sentencia ALTER TABLE en una tabla que está siendo referenciada, si se quiere modificar el esquema de la tabla, se recomienda eliminar la tabla y volverla a crear con el nuevo esquema. Cuando MySQL hace un ALTER TABLE, puede que use de manera interna un RENAME TABLE, y por lo tanto, se confundan las restricciones de clave foránea que se refieren a la tabla. Esta restricción aplica también en el caso de la sentencia CREATE INDEX, ya que MySQL la procesa como un ALTER TABLE.

Cuando se ejecute un script para cargar registros en una base de datos, es recomendable agregar las restricciones de claves foráneas vía un ALTER TABLE. De esta manera no se tiene el problema de cargar los registros en las tablas de acuerdo a un orden lógico (las tablas referenciadas deberían ir primero).

Metodología

Crear, guiado por el profesor, las tablas que se muestran en el esquema relacional de la página siguiente utilizando integridad referencial.



A continuación se muestran datos ejemplo de cada una de las tablas::

PUESTO			
	Job_ID	Job_Description	Job_Chg_Hr
+	500	Programador	\$45.38
+	501	Analista de Sistemas	\$96.75
+	502	Diseñador de Base de Datos	\$105.00
+	503	Ingeniero Eléctrico	\$84.50
+	504	Ingeniero Mecánico	\$67.90
+	505	Ingeniero Civil	\$55.78
+	506	Limpieza	\$12.00
+	507	Analista DSS	\$34.25
+	508	Diseñador de Aplicaciones	\$48.10
+	509	Contratista	\$50.50
+	510	Albañil	\$30.23
*			

EMPLEADO						
Emp_ID	Emp_Apat	Emp_Amat	Emp_name	Emp_HireDa	Job_ID	
+	101	Martinez	Flores	Ernesto	11/9/1998	502
+	102	Gomez	Morales	Rosa	7/12/1987	501
+	103	Sosa	Estrada	Pedro Alberto	12/1/1994	503
+	104	Santos	Del Rio	Sonia	12/1/1994	501
+	105	Garcia	Zamora	Guadalupe	12/1/1994	502
+	106	Plascencia	Sanchez	Eva Elvira	12/1/1994	500
+	107	Paredes	Paredes	Beatriz	12/1/1994	500
+	108	Fox	Estudillo	Valeriano	12/1/1994	501
+	109	Granados	Carballido	Julián	12/1/1994	501
+	110	Ponce	Santana	Jose	12/1/1994	505
+	111	Gonzalez	Martinez	Amanda	12/1/1994	506
+	112	Gomez	Juarez	Victoria	12/1/1994	507
+	113	Salido	Rosas	Octavio	12/1/1994	508
+	114	Macias	Montes	William	12/1/1994	508
+	115	Morín	Contreras	Christopher	12/1/1994	501
+	116	Lopez	Mendoza	Miguel	12/1/1994	510
+	117	Méndez	Trujillo	Hilda	12/1/1994	509
+	118	Morales	Castrejon	Mayra	12/1/1994	510
*						

PROYECTO	
Proy_ID	Proy_Name
+	15 Evergreen
+	18 Amber Wave
+	22 Rolling Tide
+	25 StarFlight
*	

ASIGNACION					
Asig_ID	Asig_Date	Proy_ID	Emp_ID	Asig_Hrs	A
1001	3/4/2002	15	103	2.6	
1002	5/7/2002	18	118	1.4	
1003	3/4/2002	15	101	3.6	
1004	7/15/2002	22	113	2.5	
1005	3/4/2002	15	103	1.9	
1006	9/27/2002	25	115	4.2	
1007	7/15/2002	22	105	5.2	
1008	9/27/2002	25	101	1.7	
1009	3/4/2002	15	105	2	
1010	3/4/2002	15	102	3.8	
1011	7/15/2002	22	104	2.6	
1012	3/4/2002	15	101	2.3	
1013	9/27/2002	25	114	1.8	
1014	7/15/2002	22	111	4	
1015	9/27/2002	25	114	3.4	
1016	5/7/2002	18	112	1.2	
1017	5/7/2002	18	118	2	
1018	5/7/2002	18	104	2.6	
1019	3/4/2002	15	103	3	
1020	7/15/2002	22	105	2.7	
1021	9/27/2002	25	108	4.2	
1022	9/27/2002	25	114	5.8	
1023	7/15/2002	22	106	2.4	

Tarea:

Investigar:

Cuál es el objetivo y características de las llaves primarias y foráneas.

Base de Datos

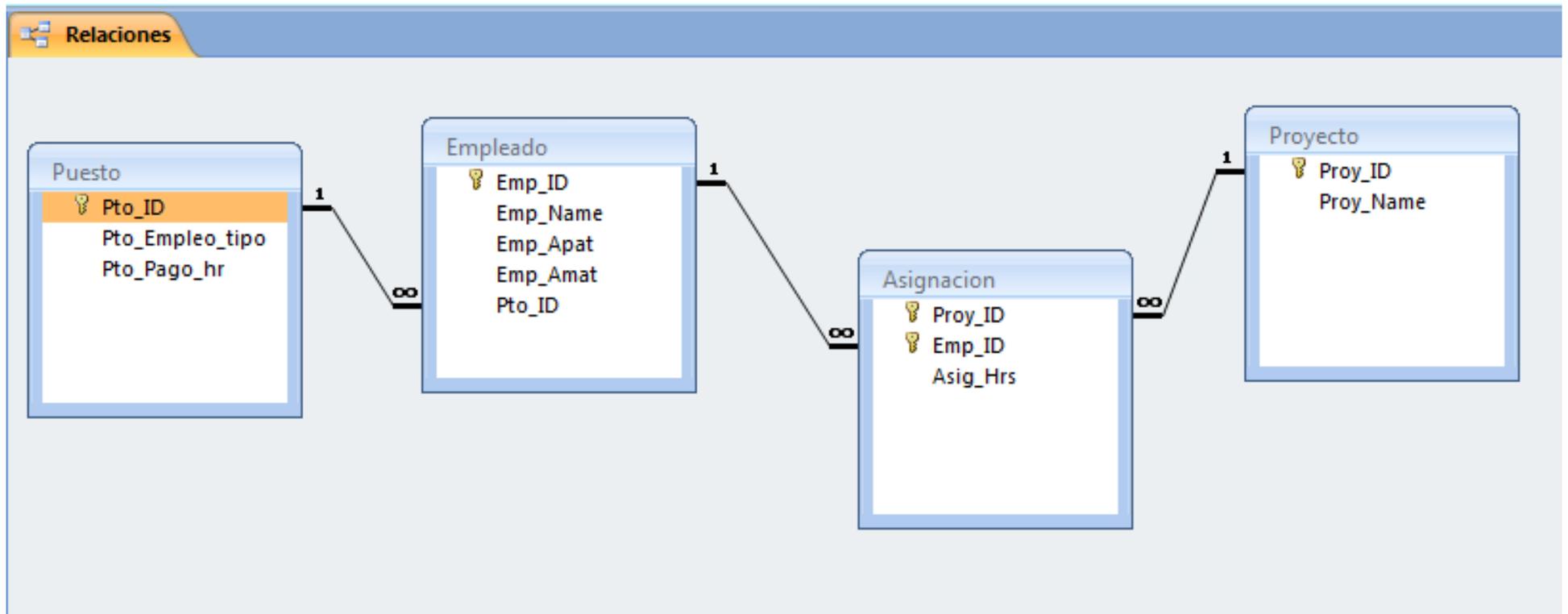
Práctica 3.

Objetivo:

Conocer cómo funciona la actualización y el borrado en cascada de la integridad referencial.

Metodología

Utilizando la base de datos CONSTRUCTORA, de la práctica 2:



Insertar los datos que se muestran a continuación, es importante recordar que hay un orden que se debe seguir al momento de la inserción de datos, el cual está relacionado con el concepto de tablas padres y tablas hijo.

Relaciones	Proyecto	Puesto	Empleado
Proy_ID	Proy_Name		
+	1	Cerro Prieto	
+	2	Laguna Ojo de Liebre	
+	3	Aguamilpa	
+	4	Peñitas	
✎	5	Puente Grande	
*			

Relaciones	Proyecto	Puesto	Empleado
Pto_ID	Pto_Empleo	Pto_Pago_h	
+	11	Arquitecto	500
+	12	Administrador	450
+	13	Contratista	460
+	14	Albañil	200
*			

Relaciones	Proyecto	Puesto	Empleado	Asignacion	
Emp_ID	Emp_Name	Emp_Apat	Emp_Amat	Pto_ID	
+	200	Melissa	Castillo	Gallardo	12
+	201	Misael	Fernández	Chong	11
+	203	Roberto	Hernández	Madrigal	11
+	204	Yuridia	Ruiz	Vargas	13
+	205	Denisse	Palleiro	Solís	12
+	206	Alberto	Armenta	Zamora	14
+	207	Andrés	Vega	Luna	14
+	208	Roberto	Pérez	Ramírez	14
+	209	Yuridia	Vega	Lafarga	13
+	210	Arturo	Santos	Lizárraga	14
*					

Asignacion	Proy_ID	Emp_ID	Asig_Hrs
	1	200	400
	1	203	700
	1	204	200
	1	206	500
	1	207	500
	1	208	350
	3	201	700
	3	205	600
	3	209	385
	3	206	400
	3	207	400
	3	208	550
	3	210	600
*			

Realizar los siguientes cambios en la tabla padre

1. Sumar 100 a cada atributo **Pto_ID**, de manera tal ahora sean: 111,112,113 y 114
2. Cambiar el nombre del proyecto 3, de Aguamilpa a Despeñadero.
3. Cambiar el número de empleado 210 a 220.
4. Borrar el empleado 220
5. Borrar el proyecto 3

Tarea:

Investigar:

****Tipos de relaciones que existen y en qué consiste cada una de ellas.**

Entregar un reporte con sus conclusiones de la práctica 3.

Base de Datos

Práctica 4.

Objetivo:

Que el alumno identifique las tablas padre e hijo, las llaves primarias y foráneas de una base de datos, mediante la implementación de una base de datos haciendo uso de la integridad referencial.

Metodología

1. En su base de datos <**almatricula**>, crear las tablas que se muestran en la figura de la siguiente página, implementando integridad referencial
 - Identifique:
 - Como se relacionan las tablas
 - La tabla padre e hijo de cada relación
 - Las llaves primarias y foráneas de cada tabla.
 - Modifique los nombres de los atributos de las tablas, según la convención definida en el curso.
 - Los nombres de las tablas serán **almatricula**Cliente, **almatricula**Factura, **almatricula**Linea, **almatricula**Producto, ejemplo al304027Cliente.
2. Insertar los datos que se indican en la figura
3. Realizar el reporte de la práctica, basándose en el reporte anexo. El reporte DEBE incluir sus conclusiones y una hoja de portada, con su nombre y el nombre de la práctica, materia y fecha..

Nota: es importante recordar que hay un orden que se debe seguir al momento de la creación de las tablas y de la inserción de datos, el cual está relacionado con el concepto de tablas padres e hijo.

Resultados:

Entregue 1 archivo:

- Reporte con sus conclusiones.

Nombre de tabla: CLIENTE
 Clave primaria: CUS_NUM
 Clave foránea: nula

Nombre de base de datos: CH2_SALE_CO

	CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE
▶	10010	Ramas	Alfred	A	615	844-2573
+	10011	Dunne	Leona	K	713	894-1238
+	10012	Smith	Kathy	W	615	894-2285
+	10013	Olowski	Paul	F	615	894-2180
+	10014	Orlando	Myron		615	222-1672
+	10015	O'Brian	Amy	B	713	442-3381
+	10016	Brown	James	G	615	297-1228
+	10017	Williams	George		615	290-2556
+	10018	Farriss	Anne	G	713	382-7185
+	10019	Smith	Olette	K	615	297-3809

Nombre de tabla: FACTURA
 Clave primaria: INV_NUMBER
 Clave foránea: CUS_CODE

	INV_NUMBER	CUS_CODE	INV_DATE
▶	1001	10014	08-Jan-02
+	1002	10011	08-Jan-02
+	1003	10012	08-Jan-02
+	1004	10011	09-Jan-02

Nombre de tabla: LINEA
 Clave primaria: INV_NUMBER + LINE_NUMBER
 Clave foránea: INV_NUMBER, PROD_CODE

	INV_NUMBER	LINE_NUMBER	PROD_CODE	LINE_UNITS	LINE_PRICE
▶	1001		1 123-21UUY	1	\$189.99
	1001		2 SRE-657UG	3	\$2.99
	1002		1 QER-34256	2	\$18.63
	1003		1 ZZX/3245Q	1	\$6.79
	1003		2 SRE-657UG	1	\$2.99
	1003		3 001278-AB	1	\$12.95
	1004		1 001278-AB	1	\$12.95
	1004		2 SRE-657UG	2	\$2.99

Nombre de tabla: PRODUCTO
 Clave primaria: PROD_CODE
 Clave foránea: nula

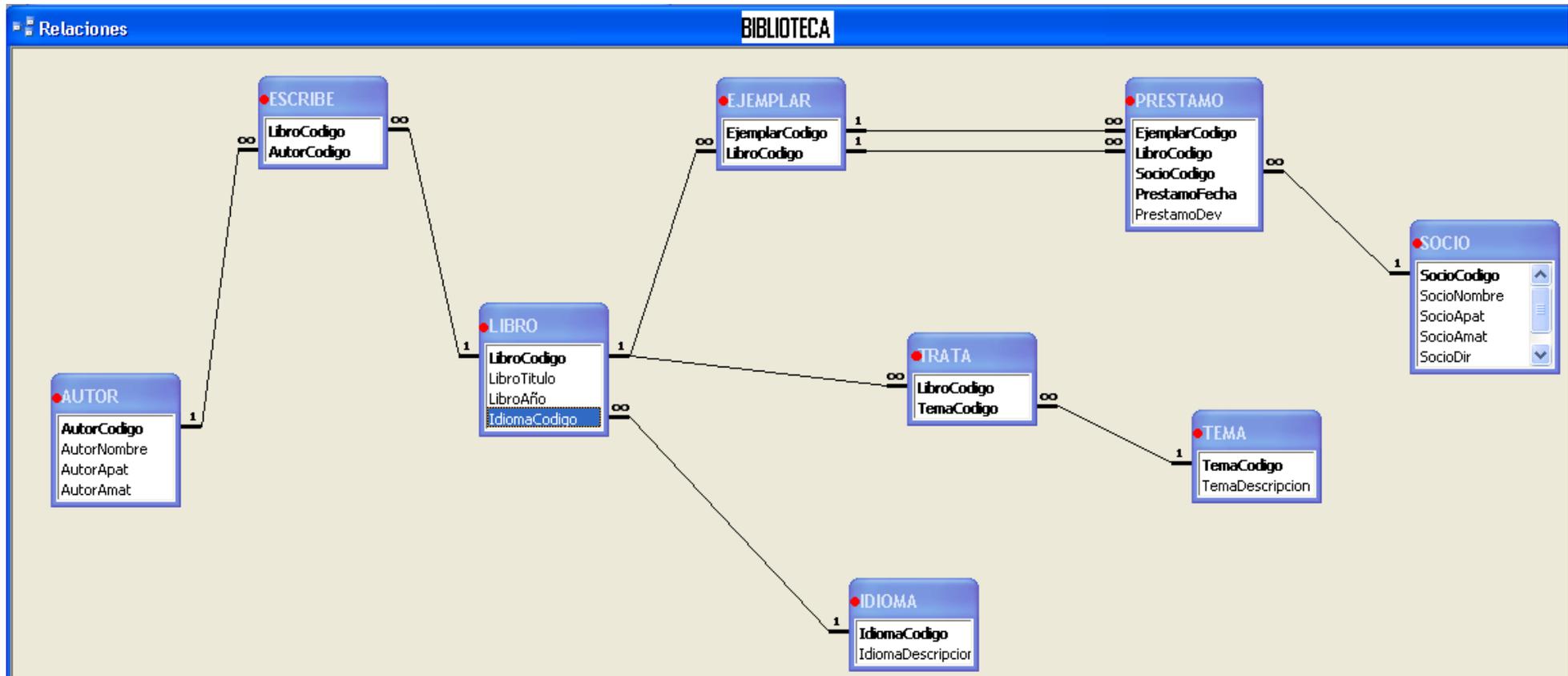
	PROD_CODE	PROD_DESCRIPTOR	PROD_PRICE	PROD_ON_HAND
▶	001278-AB	Claw hammer	\$12.95	23
+	123-21UUY	Houseite chain saw, 16-in. bar	\$189.99	4
+	QER-34256	Sledge hammer, 16-lb. head	\$18.63	6
+	SRE-657UG	Rat-tail file	\$2.99	15
+	ZZX/3245Q	Steel tape, 12-ft. length	\$6.79	8

Base de Datos

Práctica 5.

Practica de Implementación de Integridad Referencial en un manejador de base de Datos Gráfico.

Objetivo: Crear la base de Datos BIBLIOTECA en Access, con las tablas que se muestran en el siguiente esquema relacional, mediante Queries, por medio de sentencias SQL.



Nota1: Los atributos que aparecen en letras negras en cada tabla son llaves primarias, como se puede ver, en varios casos estas llaves son compuestas por más de un atributo

ESCRIBE	
LibroCodigo	int
AutorCodigo	int

int
int

AUTOR	
AutorCodigo	int
AutorNombre	Varchar(20), not null
AutorApat	Varchar(15)
AutorAmat	Varchar(15) not null

int
Varchar(20), not null
Varchar(15)
Varchar(15) not null

LIBRO	
LibroCodigo	int
LibroTitulo	Varchar(20), not null
LibroAño	int not null
IdiomaCodigo	int

int
Varchar(20), not null
int not null
int

EJEMPLAR	
EjemplarCodigo	int
LibroCodigo	int

int
int

TRATA	
LibroCodigo	int
TemaCodigo	int

int
int

IDIOMA	
IdiomaCodigo	int
IdiomaDescripcion	Varchar(15), not null

int
Varchar(15), not null

PRESTAMO	
EjemplarCodigo	int
LibroCodigo	int
SocioCodigo	int
PrestamoFecha	date
PrestamoDev	date

int
int
int
date
date

TEMA	
TemaCodigo	int
TemaDescripcion	Varchar(45), not null

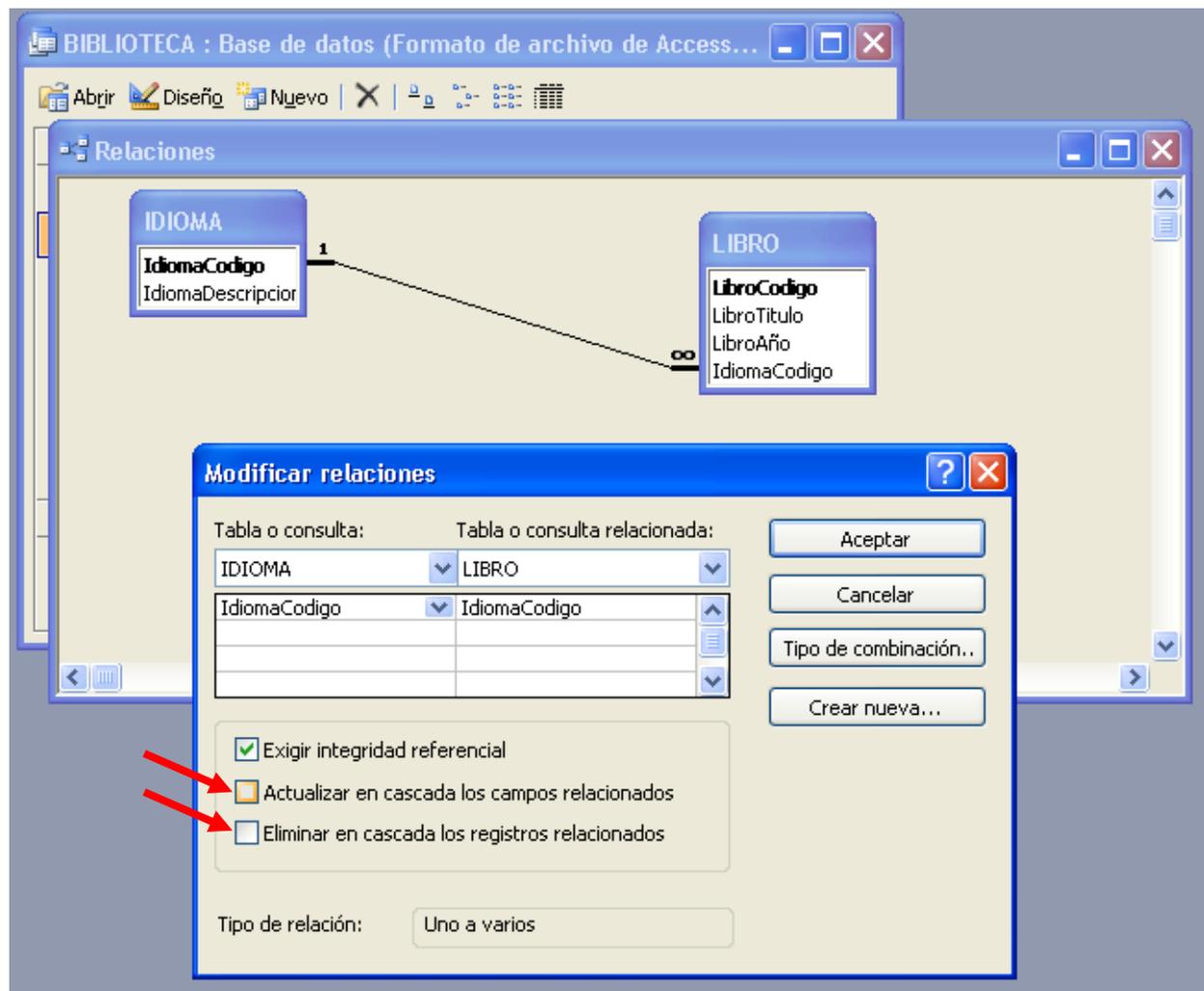
int
Varchar(45), not null

SOCIO	
SocioCodigo	int
SocioNombre	Varchar(20), not null
SocioApat	Varchar(15)
SocioAmat	Varchar(15), not null
SocioDir	Varchar(35), not null

int
Varchar(20), not null
Varchar(15)
Varchar(15), not null
Varchar(35), not null
Varchar(15)

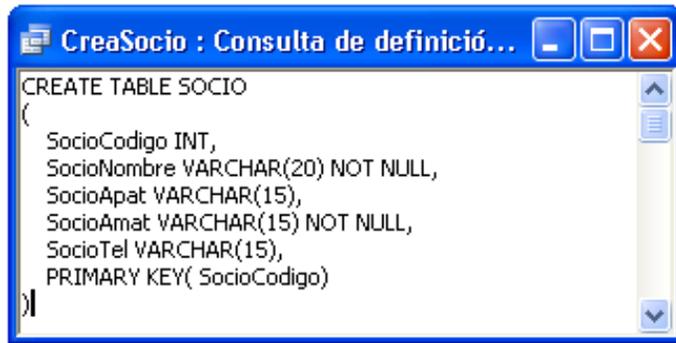
Nota2: En Access, al menos hasta la versión 2003, no se puede incluir en el Query las opciones de borrar en cascada y actualizar en cascada:
ON DELETE CASCADE,
ON UPDATE CASCADE

Por lo que, dichas opciones se deben agregar manualmente en la pantalla de relaciones como se muestra a continuación:

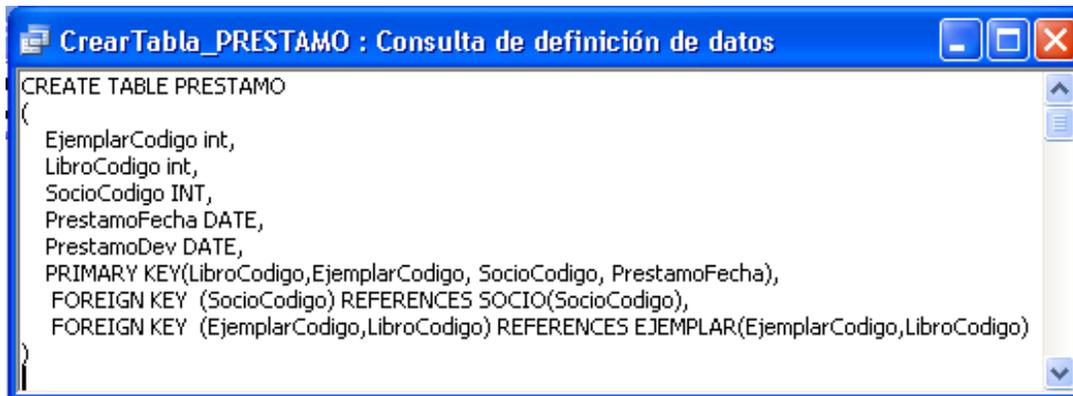


Nota3: Recuerde que debe crear las tablas con un orden de precedencia que le permita implementar la integridad referencial.

A Continuación se muestra como ejemplo el código SQL para la creación de las tablas SOCIO y PRESTAMO.



```
CREATE TABLE SOCIO
(
  SocioCodigo INT,
  SocioNombre VARCHAR(20) NOT NULL,
  SocioApat VARCHAR(15),
  SocioAmat VARCHAR(15) NOT NULL,
  SocioTel VARCHAR(15),
  PRIMARY KEY( SocioCodigo)
)
```



```
CREATE TABLE PRESTAMO
(
  EjemplarCodigo int,
  LibroCodigo int,
  SocioCodigo INT,
  PrestamoFecha DATE,
  PrestamoDev DATE,
  PRIMARY KEY(LibroCodigo,EjemplarCodigo, SocioCodigo, PrestamoFecha),
  FOREIGN KEY (SocioCodigo) REFERENCES SOCIO(SocioCodigo),
  FOREIGN KEY (EjemplarCodigo,LibroCodigo) REFERENCES EJEMPLAR(EjemplarCodigo,LibroCodigo)
)
```

Instrucciones:

Crear los Queries y copiar y pegar el código en un archivo de Word, separando con una línea de color

Cada uno de los 9 queries, además copiar la imagen de la ventana de relaciones, donde se incluyan las 9 tablas.

Base de Datos

Práctica 6.

Usando la base de datos biblioteca, realizar los queries que se especifica en cada caso, almacenarlos en la base de datos y regresar la misma con las modificaciones hechas.

Escribir los queries para realizar lo siguiente:

1. Llenar 8 filas de cada tabla
2. Modificar datos de 2 tablas diferentes, para las 8 filas de la tabla
3. Modificar datos de 3 tablas diferentes, para un individuo u objeto en específico
4. Borrar a los individuos u objetos de 2 tablas diferentes que cumplan con características determinadas por Usted.

Base de Datos

Práctica 7.

Objetivo:

Que el alumno practique la consulta y recuperación de datos en un gestor de base de datos *gráfico*.

Metodología

4. Crear la base de datos del examen práctico 1 en Access, el esquema de la base de daos esta en el manual de Mysql con clase completo, al final de la página 50.
5. Llenar al menos 3 registros en cada tabla
6. Implementar 5 diferentes búsquedas que incluyan como mínimo 3 tablas en cada búsqueda.
7. Realizar el reporte de la práctica, basándose en el reporte anexo. El reporte **DEBE** incluir sus conclusiones y una hoja de portada, con su nombre y el nombre de la práctica, materia y fecha.

Nota: es importante recordar que hay un orden que se debe seguir al momento de la creación de las tablas y de la inserción de datos, el cual está relacionado con el concepto de tablas padres e hijo.

Resultados:

Entregue 2 archivos a **ejimenez@uabc.edu.mx**:

- Reporte con sus conclusiones.
- Base de datos de Access con los queries almacenados.