

# **CREACIÓN, ACTUALIZACIÓN Y CONSULTA DE UNA BASE DE DATOS CON MYSQL**

**JULIANA ISABEL RUIZ  
CODIGO 31793228  
ADALENIZ ANDREA SANCHEZ  
CODIGO 1128418584  
RODRIGO BURITICA P.  
CODIGO 94472151**

**GRUPO COLABORATIVO 301125\_7**

**TUTOR  
HERNANDO ARBEY ROBLES**

**UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA - UNAD  
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA  
CEAD PALMIRA**

**JUNIO 2016  
INTRODUCCION**

En el presente documento se encuentran las evidencias del desarrollo grupal concerniente a lo exigido para la Fase 2 del Momento Intermedio del curso de base de Datos Avanzada. De acuerdo con los aportes individuales de cada uno de los integrantes del grupo colaborativo, se podrán encontrar las respuestas a las preguntas orientadoras, las sentencias SQL con sus respectivos resultados, de acuerdo a los requerimientos de información del problema y finalmente un Manual de creación, actualización y consulta de bases de Datos con MySQL.

## **OBJETIVOS**

- Identificar los diferentes comandos brindados por el motor mysql y su utilidad, implementando cada uno de ellos en el desarrollo de cada etapa de la practica 2
- Conocer qué y cómo podemos usar los comandos DDL y DML

## **RESPUESTAS A PREGUNTAS ORIENTADORAS**

### **PREGUNTA 1: QUÉ ES SQL Y CUÁLES SON SUS PRINCIPALES COMPONENTES**

SQL son las siglas en inglés (Structured Query Language), lo cual traducido al español es “Lenguaje de consultas estructurado”. Es un lenguaje declarativo, es decir, se escribe lo que se requiere hacer y él lo ejecuta. El SQL es un lenguaje estándar ANSI (American National Standards Institute) / ISO (International Standards Organization), declarativo e interactivo con el fin de obtener, manipular y controlar información desde una base de datos.

El SQL se compone principalmente de sentencias de Definición (aquellas con las que se crea, elimina o se modifica la estructura de una base de datos), sentencias de Manipulación (con las que se ingresan, modifican, consultan o eliminan datos, etc.) y sentencias de Control (aquellas con las que se controlan permisos, concurrencias, etc.)

### **PREGUNTA 2: QUÉ ES UNA LLAVE FORÁNEA Y CUÁL ES SU UTILIDAD**

Una llave foránea es una columna o varias columnas de una tabla, las cuales nos sirven para indicar cuál es la llave primaria de la tabla de referencia (origen). Estas columnas solo pueden tener valores existentes en la clave primaria de la tabla de origen.

La utilidad de una llave foránea es establecer una conexión con la clave primaria que se referencia, ya que mediante esta relación se valida que el valor de la columna exista en la llave primaria (tabla de referencia), para ello la columna de la llave foránea y la columna de la llave primaria deben tener el mismo tipo de dato, el mismo tamaño, sin embargo, ambas columnas pueden nombrarse de diferente forma.

### **PREGUNTA 3: QUÉ PUEDE PASAR SI NO SE DEFINE ADECUADAMENTE LA INTEGRIDAD REFERENCIAL**

La integridad referencial permite controlar que los datos que recibe una llave foránea efectivamente sean datos existentes en la llave primaria de la tabla de la cual es dependiente. De no definirse la integridad referencial, la relación puede permitir datos erróneos. Cuando en la base de datos se define inadecuadamente la integridad referencial se da paso al almacenamiento de datos basura o erróneos, se pierde la seguridad de que el dato que se está insertando exista efectivamente para esa relación.

### **PREGUNTA 4: QUÉ ES UNA LLAVE CANDIDATA Y DÉ UN EJEMPLO**

Una llave candidata son todas aquellas llaves o claves que por sus características podrían llegar a ser la llave primaria, para ello los valores que pueda tomar cada una de las columnas debe ser único.

En una tabla puede haber varias claves candidatas, esta agrupa atributos e identifica posibles Tuplas de forma unívoca y sin ambigüedades.

Un ejemplo puede ser una tabla con datos de automóviles, en los cuales el automóvil puede ser identificado de manera precisa por su número de placa o por su número de motor.

Ejemplo:

Llaves Candidatas



Nplaca	Nmotor	Marca	Modelo
CNA711	CN3487651938	Mazda	2002
LUB018	KN610000B980	Sprint	2001
JUH03B	9FSA89000A01	Honda	2007

#### PREGUNTA 5: CUÁLES SON LAS OPERACIONES QUE SE HACEN CON EL DDL

Los comandos o sentencias que se hace con DDL son:

- **CREATE:** para crear objetos en la base de datos.
- **ALTER:** modifica la estructura de la base de datos.
- **DROP:** borra objetos de la base de datos.
- **TRUNCATE:** elimina todos los registros de la tabla, incluyendo todos los espacios asignados a los registros.

#### PREGUNTA 6: QUÉ OPERACIONES SE REALIZAN CON EL DML

- **BULK INSERT:** Importa un archivo de datos en una tabla o vista
- **DELETE:** elimina una o varias filas de una tabla o vista
- **INSERT:** Inserta uno o varias filas en una tabla o vista
- **MERGE:** inserta, actualiza o elimina filas en una tabla de destino según los resultados de una combinación con una tabla de origen.
- **READTEXT:** lee los valores text, ntext, image de una columna tipo text, ntext, imagen. Inicia en una posición especificada y lee un numero de bytes especifico.
- **SELECT:** Recupera filas de datos y habilita la selección de una o varias filas o columnas de una o varias tablas.
- **UPDATE:** Actualiza los datos de una o varias filas.
- **UPDATETEXT:** actualiza un campo text, ntext o image ya existente.
- **WRITETEXT:** Permite la actualización interactiva de registro mínimo de una columna text, ntext o image. Sobre escribe completamente los datos existentes en la columna afectada.

## SENTENCIAS SQL Y RESULTADOR PARA LOS REQUERIMIENTOS DE INFORMACIÓN DEL PROBLEMA DE ESTUDIO

### Problema de estudio

El Aeropuerto de Ciudad Gótica, requiere un nuevo sistema de información para agilizar las operaciones; para ello requiere una base de datos con la información que debe manejar; esta es datos de aeronaves, que pueden ser de 3 tipos (Trasatlánticos, Locales y privados), de estas aeronaves, se maneja un registro de mantenimientos que pueden ser preventivos previamente programados y correctivos cuando los requieren, también se manejan datos de Aeropuertos de destino, datos de pilotos, auxiliares de vuelo y personal administrativo, tanto pilotos como auxiliares, pueden cambiar de avión, durante uno o varios itinerarios; el sistema debe permitir crear itinerarios de vuelos que incluyan Aeronave, Hora de salida, hora de llegada, aeropuerto destino y personal de abordaje, esto es piloto, copiloto y auxiliares.

Entre la información que el sistema debe arrojar esta:

1. Lista general de Aeronaves.
2. Lista de pilotos.
3. Lista de auxiliares.
4. Lista de personal administrativo.
5. Lista de pilotos, por tipo de aeronave que pueden manejar
6. Lista de auxiliares según su antigüedad en la empresa
7. Reporte de la información de un funcionario a partir de su cédula
8. Reporte de un funcionario, según el apellido o el nombre cuando no se conoce la cédula
9. Relación de pilotos y/o auxiliares y aeronaves en las que trabaja.
10. Lista de pilotos que manejan un determinado avión.
11. Lista de aviones que maneja un piloto
12. Lista de aeropuertos a los que llegan la aeronaves.
13. Relación de itinerarios que se tienen en un rango de fechas.
14. Relación de pilotos que participan en los itinerarios de un rango de fechas.
15. Relación de itinerarios en los que participa un piloto.
16. Relación de pilotos y auxiliares vinculados a un itinerario.

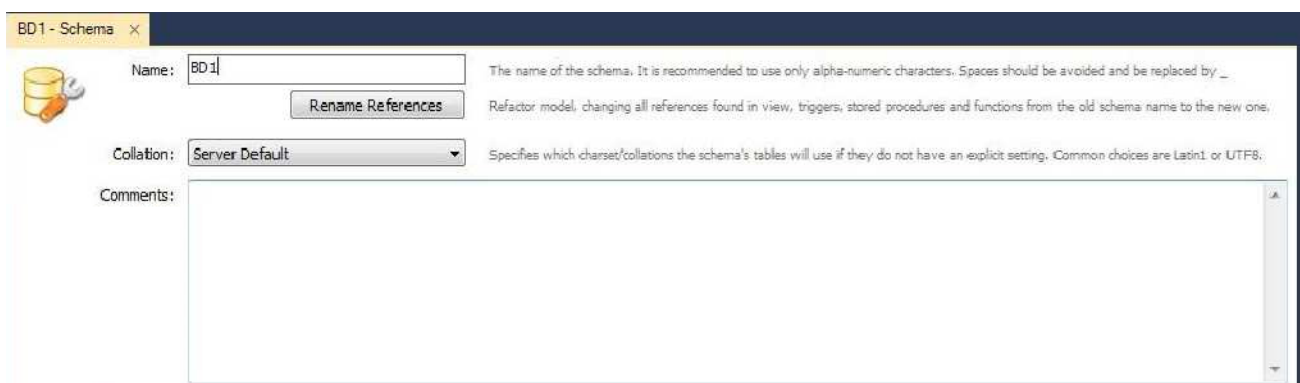
17. lista de vuelos por fecha.
18. Lista de pilotos y auxiliares por fecha.
19. Lista de pilotos y auxiliares que han faltado en un rango de fechas.
20. Lista de aeronaves ordenadas por fecha de entrada en operación.
21. Lista de personal administrativo, ordenado por cargo y antigüedad.
22. Relación de mantenimientos programados en un rango de fechas.
23. Relación de mantenimientos preventivos de una determinada aeronave.
24. Relación de mantenimientos correctivos en un rango de fechas.
25. Relación de mantenimientos correctivos a una aeronave, en un rango de fechas.

## CREACIÓN DE LA BASE DE DATOS DEL AEROPUERTO

Abrimos la herramienta Mysql Workbench



Le damos clic en el + que aparece en la parte derecha allí abre un nuevo schema que nos generara la Base de Datos, le damos un nombre.



## SENTENCIAS DE CREACIÓN DE LA BASE DE DATOS

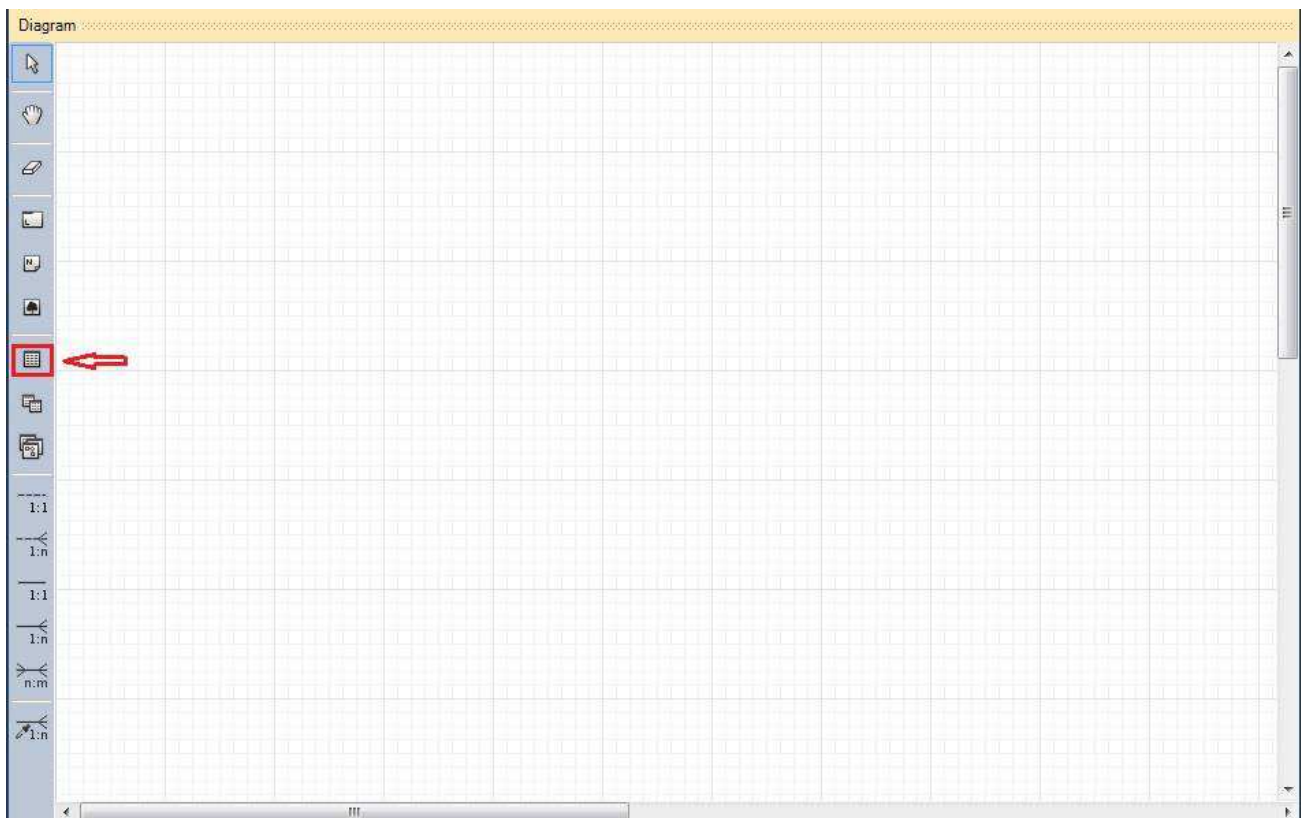
Para el presente documento, tomaremos la base de datos de la compañera Juliana Ruíz, creada con el nombre JRuiz

CREATE DATABASE JRuiz;

Creada ya la Base de datos, iniciamos con la creación de las tablas de forma Gráfica, damos clic en Add Diagram:



Aquí arrastramos hasta el área del diagrama, le damos el nombre a cada tabla, sus respectivos atributos, llaves primarias y foráneas.



- **TABLA AERONAVES**



aeronaves - Table										
Table Name:									Schema:	
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
cod_aeronaves	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
descripcion	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_marca	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
piloto	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
copiloto	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
auxiliar	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
cod_mantenimiento	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-- Table `JRuiz`.`aeronaves`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`aeronaves` ( `cod_aeronaves` INT NOT NULL AUTO_INCREMENT COMMENT "",
`descripcion` VARCHAR(45) NOT NULL COMMENT "", `id_marca` INT NOT NULL COMMENT "", `piloto` INT NOT NULL
COMMENT "", `copiloto` INT NOT NULL COMMENT "", `auxiliar` INT NOT NULL COMMENT "", `cod_mantenimiento` INT
NOT NULL COMMENT "", PRIMARY KEY (`cod_aeronaves`) COMMENT "",
CONSTRAINT `mantenimiento`
FOREIGN KEY (`cod_mantenimiento`)
REFERENCES `JRuiz`.`mantenimiento` (`cod_mantenimiento`)
ON DELETE NO ACTION
ON UPDATE NO ACTION, CONSTRAINT `marca`
FOREIGN KEY (`id_marca`)
REFERENCES `JRuiz`.`marca` (`id_marca`)
ON DELETE NO ACTION
ON UPDATE NO ACTION, CONSTRAINT `piloto`
FOREIGN KEY (`piloto`)
REFERENCES `JRuiz`.`funcionarios` (`id_funcionarios`)
ON DELETE NO ACTION
ON UPDATE NO ACTION, CONSTRAINT `copiloto`
FOREIGN KEY (`copiloto`)
REFERENCES `JRuiz`.`funcionarios` (`id_funcionarios`)
ON DELETE NO ACTION
ON UPDATE NO ACTION, CONSTRAINT `auxiliar`
FOREIGN KEY (`auxiliar`)
REFERENCES `JRuiz`.`funcionarios` (`id_funcionarios`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE INDEX `mantenimiento_idx` ON `JRuiz`.`aeronaves` (`cod_mantenimiento` ASC) COMMENT "";
CREATE INDEX `marca_idx` ON `JRuiz`.`aeronaves` (`id_marca` ASC) COMMENT "";
CREATE INDEX `piloto_idx` ON `JRuiz`.`aeronaves` (`piloto` ASC) COMMENT "";
CREATE INDEX `copiloto_idx` ON `JRuiz`.`aeronaves` (`copiloto` ASC) COMMENT "";
CREATE INDEX `auxiliar_idx` ON `JRuiz`.`aeronaves` (`auxiliar` ASC) COMMENT "";
```

- **TABLA AEROPUERTOS**

aeropuertos - Table										
Table Name: <input type="text" value="aeropuertos"/>										Schema:
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
cod_aeropuertos	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
descripcion	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

### Sentencias

```
-- Table `JRuiz`.`aeropuertos`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`aeropuertos` ( `cod_aeropuertos` INT NOT NULL AUTO_INCREMENT COMMENT "", `descripcion` VARCHAR(100) NOT NULL COMMENT "", PRIMARY KEY (`cod_aeropuertos`) COMMENT "") ENGINE = InnoDB;
```

- **TABLA CARGO**

cargo - Table										
Table Name: <input type="text" value="cargo"/>										Schema:
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
id_cargo	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
descripcion	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

### Sentencias

```
-- Table `JRuiz`.`cargo`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`cargo` ( `id_cargo` INT NOT NULL AUTO_INCREMENT COMMENT "", `descripcion` VARCHAR(50) NOT NULL COMMENT "", PRIMARY KEY (`id_cargo`) COMMENT "") ENGINE = InnoDB;
```

- **TABLA CIUDAD**

ciudad Table										
Table Name: ciudad		Schema:								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
id_ciudad	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
ciudad	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_departamento	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
cod aeropuertos	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-- Table `JRuiz`.`ciudad`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`ciudad` ( `id_ciudad` INT NOT NULL AUTO_INCREMENT COMMENT "", `ciudad` VARCHAR(45) NOT NULL COMMENT "", `id_departamento` INT NOT NULL COMMENT "", `cod_aeropuertos` INT NULL COMMENT "", PRIMARY KEY (`id_ciudad`) COMMENT "", CONSTRAINT `id_departamento` FOREIGN KEY (`id_departamento`) REFERENCES `JRuiz`.`departamento` (`id_departamento`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `cod_aeropuertos` FOREIGN KEY (`cod_aeropuertos`) REFERENCES `JRuiz`.`aeropuertos` (`cod_aeropuertos`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE = InnoDB;
CREATE INDEX `id_departamento_idx` ON `JRuiz`.`ciudad` (`id_departamento` ASC) COMMENT "";
CREATE INDEX `cod_aeropuertos_idx` ON `JRuiz`.`ciudad` (`cod_aeropuertos` ASC) COMMENT "";
```

## • TABLA DEPARTAMENTO

departamento - Table										
Table Name: departamento		Schema:								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
id_departamento	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
departamento	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_nacionalidad	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-----
-- Table `JRuiz`.`departamento`
-----
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`departamento` ( `id_departamento` INT NOT NULL AUTO_INCREMENT
COMMENT "`, `departamento` VARCHAR(45) NOT NULL COMMENT "`, `id_nacionalidad` INT NOT NULL COMMENT "`,
PRIMARY KEY (`id_departamento`) COMMENT "`, CONSTRAINT `id_nacionalidad`
FOREIGN KEY (`id_nacionalidad`)
REFERENCES `JRuiz`.`nacionalidad` (`id_nacionalidad`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE INDEX `id_nacionalidad_idx` ON `JRuiz`.`departamento` (`id_nacionalidad` ASC) COMMENT "`,`
```

## • TABLA ESTADO\_VUELO

estado_vuelo - Table										
Table Name: estado_vuelo		Schema:								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
cod_estado_vuelo	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
descripcion	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-----
-- Table `JRuiz`.`estado_vuelo`
-----
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`estado_vuelo` ( `cod_estado_vuelo` INT NOT NULL AUTO_INCREMENT
COMMENT "`, `descripcion` VARCHAR(45) NOT NULL COMMENT "`, PRIMARY KEY (`cod_estado_vuelo`) COMMENT "`)
ENGINE = InnoDB;
```

## • TABLA FUNCIONARIOS

funcionarios - Table										
Table Name: funcionarios		Schema:								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
id_funcionarios	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
id_personas	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_cargo	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-----
-- Table `JRuiz`.`funcionarios`
-----
```

```

CREATE TABLE IF NOT EXISTS `JRuiz`.`funcionarios` ( `id_funcionarios` INT NOT NULL AUTO_INCREMENT COMMENT
", `id_personas` INT NOT NULL COMMENT ", `id_cargo` INT NOT NULL COMMENT ", PRIMARY KEY (`id_funcionarios`)
COMMENT ", CONSTRAINT `id_cargo`
FOREIGN KEY (`id_cargo`)
REFERENCES `JRuiz`.`cargo` (`id_cargo`)
ON DELETE NO ACTION
ON UPDATE NO ACTION, CONSTRAINT `d_personas`
FOREIGN KEY (`id_personas`)
REFERENCES `JRuiz`.`personas` (`id_personas`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE INDEX `id_cargo_idx` ON `JRuiz`.`funcionarios` (`id_cargo` ASC) COMMENT ";
CREATE INDEX `d_personas_idx` ON `JRuiz`.`funcionarios` (`id_personas` ASC) COMMENT ";

```

- **TABLA GENERO**

genero - Table									
Table Name: <input type="text" value="genero"/>		Schema:							
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id_genero	CHAR(1)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
descripcion	VARCHAR(9)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

## Sentencias

```

-----
-- Table `JRuiz`.`genero`
-----
CREATE TABLE IF NOT EXISTS `JRuiz`.`genero` ( `id_genero` CHAR(1) NOT NULL COMMENT ", `descripcion`
VARCHAR(9) NOT NULL COMMENT ", PRIMARY KEY (`id_genero`) COMMENT ")
ENGINE = InnoDB;

```

- **TABLA INTINERARIOS**

itinerarios - Table										
Table Name: <input type="text" value="itinerarios"/>		Schema:								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
cod_itinerarios	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
destino	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
tarifa	DECIMAL(8)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-- Table `JRuiz`.`itinerarios`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`itinerarios` ( `cod_itinerarios` INT NOT NULL AUTO_INCREMENT COMMENT "",
`destino` INT NOT NULL COMMENT "", `tarifa` DECIMAL(8) NOT NULL COMMENT "", PRIMARY KEY (`cod_itinerarios`)
COMMENT "", CONSTRAINT `ciudad`
FOREIGN KEY (`destino`)
REFERENCES `JRuiz`.`ciudad` (`id_ciudad`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE INDEX `ciudad_idx` ON `JRuiz`.`itinerarios` (`destino` ASC) COMMENT "";
```

## • TABLA MANTENIMIENTO

mantenimiento - Table										
Table Name: <input type="text" value="mantenimiento"/>		Schema:								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
cod_mantenimiento	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
descripcion	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
fecha_mantenimiento	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-- Table `JRuiz`.`mantenimiento`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`mantenimiento` ( `cod_mantenimiento` INT NOT NULL AUTO_INCREMENT
COMMENT "", `descripcion` VARCHAR(45) NOT NULL COMMENT "", `fecha_mantenimiento` DATETIME NOT NULL
COMMENT "", PRIMARY KEY (`cod_mantenimiento`) COMMENT "")
ENGINE = InnoDB;
```

- **TABLA MARCA**

marca - Table										
Table Name: <input type="text" value="marca"/>		Schema:								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
id_marca	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
nom_marca	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-- Table `JRuiz`.`marca`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`marca` ( `id_marca` INT NOT NULL AUTO_INCREMENT COMMENT "",
`nom_marca` VARCHAR(45) NOT NULL COMMENT "", PRIMARY KEY (`id_marca`) COMMENT "")
ENGINE = InnoDB;
```

- **TABLA NACIONALIDAD**

nacionalidad - Table										
Table Name: <input type="text" value="nacionalidad"/>		Schema:								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
id_nacionalidad	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
nacionalidad	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-- Table `JRuiz`.`nacionalidad`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`nacionalidad` ( `id_nacionalidad` INT NOT NULL AUTO_INCREMENT
COMMENT "", `nacionalidad` VARCHAR(45) NOT NULL COMMENT "", PRIMARY KEY (`id_nacionalidad`) COMMENT "")
ENGINE = InnoDB;
```

- **TABLA PERSONAS**



personas - Table										
Table Name: <input type="text" value="personas"/>		Schema: <input type="text"/>								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
id_personas	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
doc_personas	VARCHAR(12)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_tipo_documento	CHAR(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
nom_persona	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
apell_persona	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
fecha_nacimiento	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
fecha_ingreso	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_genero	CHAR(2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
celular	VARCHAR(12)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
id_ciudad	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
email	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-- Table `JRuiz`.`personas`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`personas` ( `id_personas` INT NOT NULL AUTO_INCREMENT COMMENT "
`, `doc_personas` VARCHAR(12) NOT NULL COMMENT "`, `id_tipo_documento` CHAR(2) NOT NULL COMMENT "
`, `nom_persona` VARCHAR(45) NOT NULL COMMENT "`, `apell_persona` VARCHAR(45) NOT NULL COMMENT "
`, `fecha_nacimiento` DATE NOT NULL COMMENT "`, `fecha_ingreso` DATE NOT NULL COMMENT "`, `id_genero` CHAR(2)
NOT NULL COMMENT "`, `celular` VARCHAR(12) NOT NULL COMMENT "`, `id_ciudad` INT NOT NULL COMMENT "
`, `email` VARCHAR(100) NOT NULL COMMENT "`, PRIMARY KEY (`id_personas`) COMMENT "`, CONSTRAINT `id_ciudad`
FOREIGN KEY (`id_ciudad`)
REFERENCES `JRuiz`.`ciudad` (`id_ciudad`)
ON DELETE NO ACTION
ON UPDATE NO ACTION, CONSTRAINT `id_genero`
FOREIGN KEY (`id_genero`)
REFERENCES `JRuiz`.`genero` (`id_genero`)
ON DELETE NO ACTION
ON UPDATE NO ACTION, CONSTRAINT `id_tipo_documento`
FOREIGN KEY (`id_tipo_documento`)
REFERENCES `JRuiz`.`tipo_documento` (`id_tipo_documento`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE INDEX `id_ciudad_idx` ON `JRuiz`.`personas` (`id_ciudad` ASC) COMMENT "
;
CREATE INDEX `id_genero_idx` ON `JRuiz`.`personas` (`id_genero` ASC) COMMENT "
;
CREATE INDEX `id_tipo_documento_idx` ON `JRuiz`.`personas` (`id_tipo_documento` ASC) COMMENT "
;
```

- **TABLA TIPO\_DOCUMENTO**



tipo_documento - Table										
Table Name: <input type="text" value="tipo_documento"/>		Schema: <input type="text"/>								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
id_tipo_documento	CHAR(2)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
descripcion	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-- Table `JRuiz`.`tipo_documento`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`tipo_documento` ( `id_tipo_documento` CHAR(2) NOT NULL COMMENT "",
`descripcion` VARCHAR(45) NOT NULL COMMENT "", PRIMARY KEY (`id_tipo_documento`) COMMENT "")
ENGINE = InnoDB;
```

## • TABLA VUELOS

vuelos - Table										
Table Name: <input type="text" value="vuelos"/>		Schema: <input type="text"/>								
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
cod_vuelos	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
cod_aeronaves	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
cod_itinerario	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
cod_estado_vuelo	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
fecha_salida	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
fecha_llegada	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

## Sentencias

```
-- Table `JRuiz`.`vuelos`
```

```
CREATE TABLE IF NOT EXISTS `JRuiz`.`vuelos` ( `cod_vuelos` INT NOT NULL AUTO_INCREMENT COMMENT "",
`cod_aeronaves` INT NOT NULL COMMENT "", `cod_itinerario` INT NOT NULL COMMENT "", `cod_estado_vuelo` INT NOT
```

```

NULL COMMENT ", `fecha_salida` DATETIME NOT NULL COMMENT ", `fecha_llegada` DATETIME NOT NULL
COMMENT ", PRIMARY KEY (`cod_vuelos`) COMMENT ", CONSTRAINT `aeronaves`
FOREIGN KEY (`cod_aeronaves`)
REFERENCES `JRuiz`.`aeronaves` (`cod_aeronaves`)
ON DELETE NO ACTION
ON UPDATE NO ACTION, CONSTRAINT `itinerario`
FOREIGN KEY (`cod_itinerario`)
REFERENCES `JRuiz`.`itinerarios` (`cod_itinerarios`)
ON DELETE NO ACTION
ON UPDATE NO ACTION, CONSTRAINT `vuelos`
FOREIGN KEY (`cod_estado_vuelo`)
REFERENCES `JRuiz`.`estado_vuelo` (`cod_estado_vuelo`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE INDEX `itinerario_idx` ON `JRuiz`.`vuelos` (`cod_itinerario` ASC) COMMENT ";
CREATE INDEX `vuelos_idx` ON `JRuiz`.`vuelos` (`cod_estado_vuelo` ASC) COMMENT ";
CREATE INDEX `aeronaves_idx` ON `JRuiz`.`vuelos` (`cod_aeronaves` ASC) COMMENT ";

```

## CONSULTAS

De acuerdo con los requerimientos del problema de Estudio, las consultas a realizar sobre la base de datos del Aeropuerto son las siguientes:

- LISTA GENERAL DE AERONAVES

**SELECT \* FROM aeronaves;**

	cod_aeronaves	descripcion	id_marca	piloto	copiloto	auxiliar	cod_mantenimiento
▶	1	Trasatlanticos	4	4	2	2	1
	2	Locales	2	5	2	3	2
	3	Privados	1	4	2	3	3
	4	Locales	2	5	2	2	4
	5	Privados	1	4	2	2	5
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

aeronaves 1 x

- REPORTE DE LA INFORMACIÓN DE UN FUNCIONARIO A PARTIR DE SU CEDULA

**SELECT \* FROM personas**  
**WHERE doc\_personas='31793228'**

id_personas	doc_personas	id_tipo_documento	nom_persona	apell_persona	fecha_nacimiento	fecha_ingreso	id_genero	celular	id_ciudad	email
1	31793228	CC	Juliana	Ruiz	1981-07-21	2014-10-10	F	3117970089	1	sistemas.sees@gmail.com
* NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- REPORTE DE UN FUNCIONARIO SEGÚN EL APELLIDO O EL NOMBRE CUANDO NO SE CONOCE LA CÉDULA

```
SELECT * FROM personas
WHERE nom_persona ='Julio Cesar'
```

id_personas	doc_personas	id_tipo_documento	nom_persona	apell_persona	fecha_nacimiento	fecha_ingreso	id_genero	celular	id_ciudad	email
4	1116235678	CC	Julio Cesar	Ruiz	1985-09-25	2012-12-01	M	3229998983	1	julioruiz@gmail.com
* NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- LISTA DE PILOTOS

```
SELECT id_cargo, nom_persona, descripcion FROM cargo, personas;
WHERE id_cargo ='1'
```

id_cargo	nom_persona	descripcion
1	Juliana	Piloto
1	Colombia	Piloto
1	Xiomra	Piloto
1	Julio Cesar	Piloto
1	Juan Sebastian	Piloto

- LISTA DE AUXILIARES

**SELECT** id\_cargo, nom\_persona, descripcion **FROM** cargo, personas;  
**WHERE** id\_cargo ='3'

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
id_cargo	nom_persona	descripcion	
3	Juliana	Auxiliar	
3	Colombia	Auxiliar	
3	Xiomra	Auxiliar	
3	Julio Cesar	Auxiliar	
3	Juan Sebastian	Auxiliar	

- **LISTA DE PERSONAL ADMINISTRATIVO**

**SELECT** id\_cargo, nom\_persona, descripcion **FROM** cargo, personas;  
**WHERE** id\_cargo ='4'

Result Grid	Filter Rows:	Exports:	Wrap Cell Content:
id_cargo	nom_persona	descripcion	
4	Juliana	Administrativo	
4	Colombia	Administrativo	
4	Xiomra	Administrativo	
4	Julio Cesar	Administrativo	
4	Juan Sebastian	Administrativo	

- **RELACIÓN DE PILOTOS Y/O AUXILIARES Y AERONAVES EN LAS QUE TRABAJA**

**SELECT** piloto, auxiliar, descripcion  
**FROM** aeronaves, personas;

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
piloto	auxiliar	descripcion	
1	3	Trasatlanticos	
1	1	PRIVADOS	
2	2	LOCALES	
1	3	Trasatlanticos	
1	1	PRIVADOS	
2	2	LOCALES	

- **LISTA DE PILOTOS QUE MANEJAN UN DETERMINADO AVION**

## SELECT

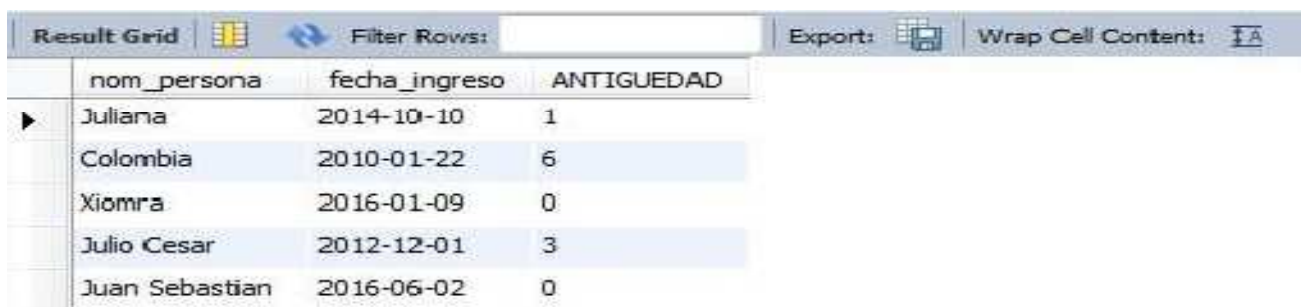
```
cod_aeronaves, descripcion, id_marca, piloto, copiloto, auxiliar, cod_mantenimiento
FROM aeronaves
WHERE descripcion = 'LOCALES'
```



	cod_aeronaves	descripcion	id_marca	piloto	copiloto	auxiliar	cod_mantenimiento
▶	3	LOCALES	3	2	1	2	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- LISTA DE PERSONAL ADMINISTRATIVO ORDENADO POR CARGO Y ANTIGUEDAD

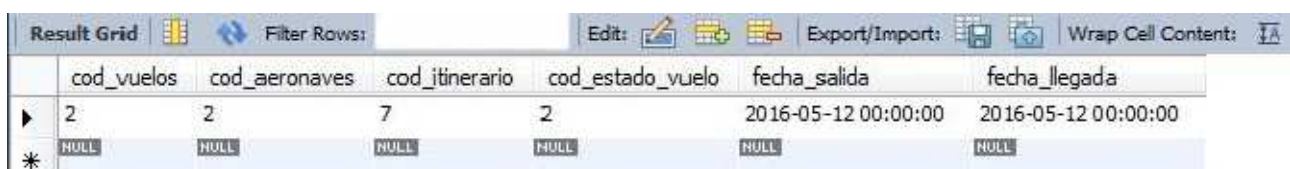
```
SELECT nom_persona, fecha_ingreso,
TIMESTAMPDIFF(YEAR, fecha_ingreso, CURDATE()) AS ANTIGUEDAD
FROM personas;
```



	nom_persona	fecha_ingreso	ANTIGUEDAD
▶	Juliana	2014-10-10	1
	Colombia	2010-01-22	6
	Xiomra	2016-01-09	0
	Julio Cesar	2012-12-01	3
	Juan Sebastian	2016-06-02	0

- LISTA DE VUELOS POR FECHA

```
SELECT * FROM vuelos
WHERE fecha_salida = "2016-05-12"
```



	cod_vuelos	cod_aeronaves	cod_itinerario	cod_estado_vuelo	fecha_salida	fecha_llegada
▶	2	2	7	2	2016-05-12 00:00:00	2016-05-12 00:00:00
*	NULL	NULL	NULL	NULL	NULL	NULL

- LISTA DE AERONAVES ORDENADAS POR FECHA DE ENTRADA EN OPERACIÓN

```
SELECT * FROM aeronaves
ORDER BY fecha_operacion ASC;
```

	cod_aeronaves	descripcion	id_marca	piloto	copiloto	auxiliar	cod_mantenimiento	cod_aeropuertos	fecha_operacion
1	1	Trasatlanticos	1	1	2	3	1	1	2016-01-11
2	2	PRIVADOS	2	1	1	1	2	2	2016-01-12
3	3	LOCALES	3	2	1	2	1	3	2016-01-15

- **LISTA DE PILOTOS Y AUXILIARES QUE HAN OPERADO EN UN RANGO DE FECHAS**

```
SELECT piloto, auxiliar
FROM aeronaves
WHERE fecha_operacion >= "2015-05-12" AND fecha_operacion < "2016-01-11";
```

	piloto	auxiliar	fecha_operacion
--	--------	----------	-----------------

- **LISTA DE PILOTOS Y AUXILIARES EN UNA FECHA DE OPERACION**

```
SELECT piloto, auxiliar
FROM aeronaves
WHERE fecha_operacion = "2016-01-11";
```

	piloto	auxiliar
1	1	3

- **LISTA DE AUXILIARES SEGÚN SU ANTIGUEDAD EN LA EMPRESA**

```
SELECT auxiliar, nom_persona,
TIMESTAMPDIFF(YEAR, fecha_ingreso, CURDATE()) AS ANTIGUEDAD
FROM personas, aeronaves;
ORDER BY ANTIGUEDAD ASC;
```

	auxiliar	nom_persona	ANTIGUEDAD
1	1	Juliana	1
2	2	Juliana	1
3	3	Juliana	1
1	1	Colombia	6
2	2	Colombia	6
3	3	Colombia	6



- **RELACION DE ITINERARIOS EN UN RANGO DE FECHAS**

```
SELECT *
FROM itinerarios
WHERE fecha ="2016-06-02";
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
cod_itinerarios	destino	tarifa	fecha	
8	3	89000	2016-06-02	
* NULL	NULL	NULL	NULL	

## ACTUALIZACIÓN

Las siguientes fueron actualizaciones realizadas sobre la base de datos del Aeropuerto

- **A la Aeronave No 1 se le actualiza la descripción por TRASATLÁNTICO**

```
UPDATE aeronaves
SET
descripcion = 'TRASATLANTICO'
WHERE cod_aeronaves = 1
```

Result Grid										Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
	cod_aeronaves	descripcion	id_marca	piloto	copiloto	auxiliar	cod_mantenimiento	cod_aeropuertos	fecha_operacion								
▶	1	TRASATLANTICO	1	1	2	3	1	1	2016-01-11								
	2	PRIVADOS	2	1	1	1	2	2	2016-01-12								
	3	LOCALES	3	2	1	2	1	3	2016-01-15								
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL								

- **Al Aeropuerto No 5 se le cambia la descripción por Ernesto Cortissoz**

```
UPDATE aeropuertos
SET
descripcion='Ernesto Cortissoz'
WHERE cod_aeropuertos=5;
```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	cod_aeropuertos	descripcion			
▶	1	Bonilla Aragon			
	2	El Dorado			
	3	Jose Maria Cordoba			
	4	El eden			
	5	Ernesto Cortissoz			

- Cambio del nombre y los apellidos de la persona No 3 por Xiomara Villamil Triana

**UPDATE** personas

**SET**

nom\_persona = 'Xiomara', apell\_persona = 'Villamil Triana'

**WHERE** id\_personas = '3';

Result Grid		Filter Rows:		Edit:	Export/Import:		Wrap Cell Content:		
	id_personas	doc_personas	id_tipo_documento	nom_persona	apell_persona	fecha_nacimiento	fecha_ingreso	id_genero	celular
▶	1	31793228	CC	Juliana	Ruiz	1981-07-21	2014-10-10	F	3117970089
	2	31190057	CC	Colombia	Osorio	1953-04-07	2013-01-22	F	3116789909
	3	52276890	CC	Xiomara	Villamil Triana	1976-08-07	2015-01-09	F	3108140505
	4	1116235678	CC	Julio Cesar	Ruiz	1985-09-25	2012-12-01	M	3229998983



# MANUAL DE CREACIÓN, ACTUALIZACIÓN Y CONSULTA DE BASES DE DATOS CON MySQL

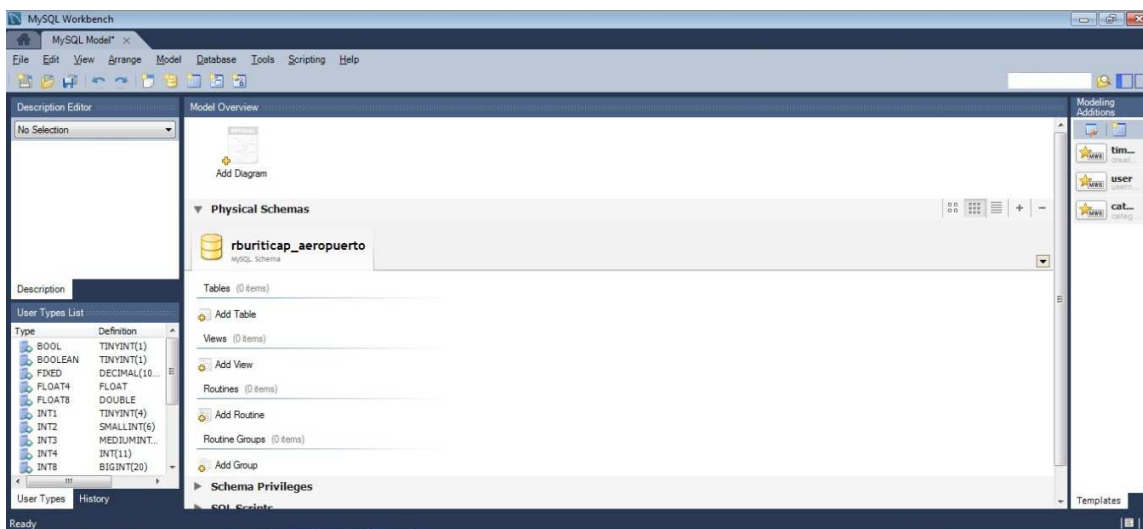
El siguiente y sencillo manual explica de manera gráfica los procesos de creación de una BD a partir de un modelo creado en MySQL Workbench, inserción de datos, actualización, consultas y eliminación de registros. Este manual toma como ejemplo la base de datos del Aeropuerto.

## CREACIÓN DE LA BASE DE DATOS

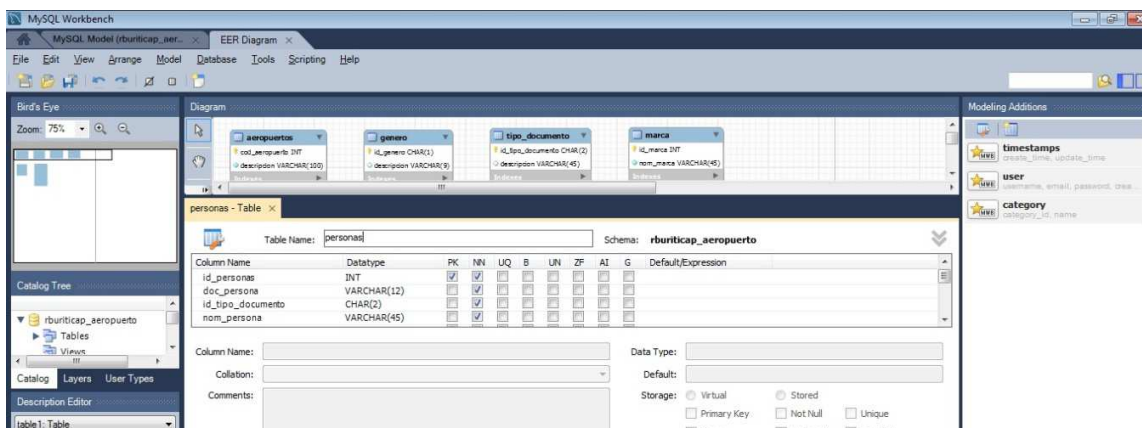
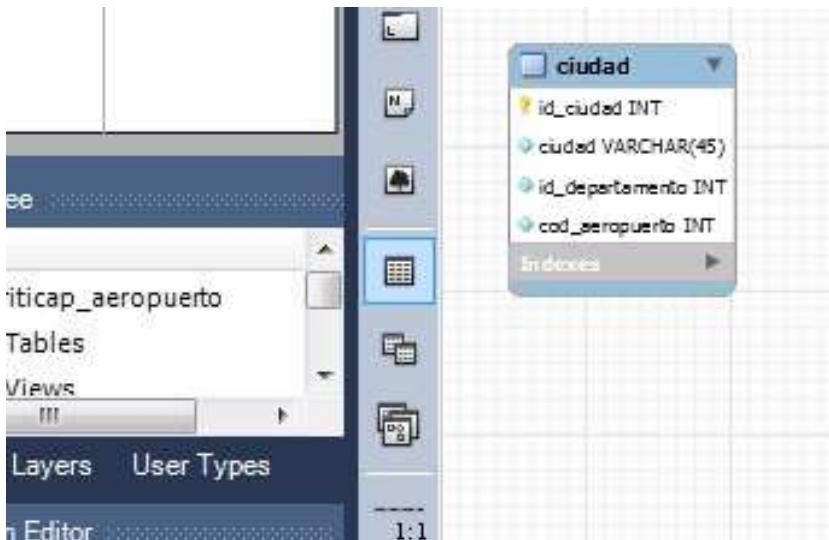
- Iniciamos el servidor xampp



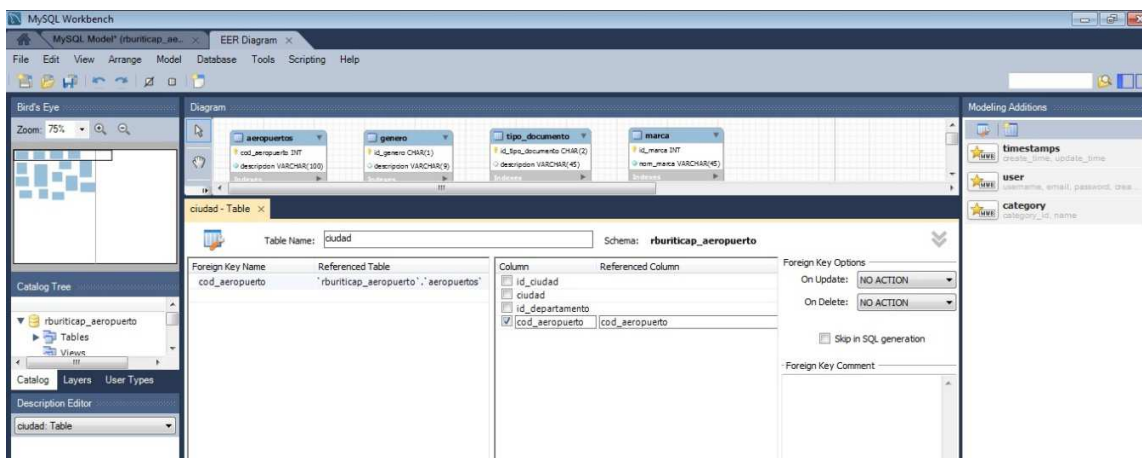
- Entramos a MySQL Workbench y creamos el nuevo modelo rburiticap\_aeropuerto



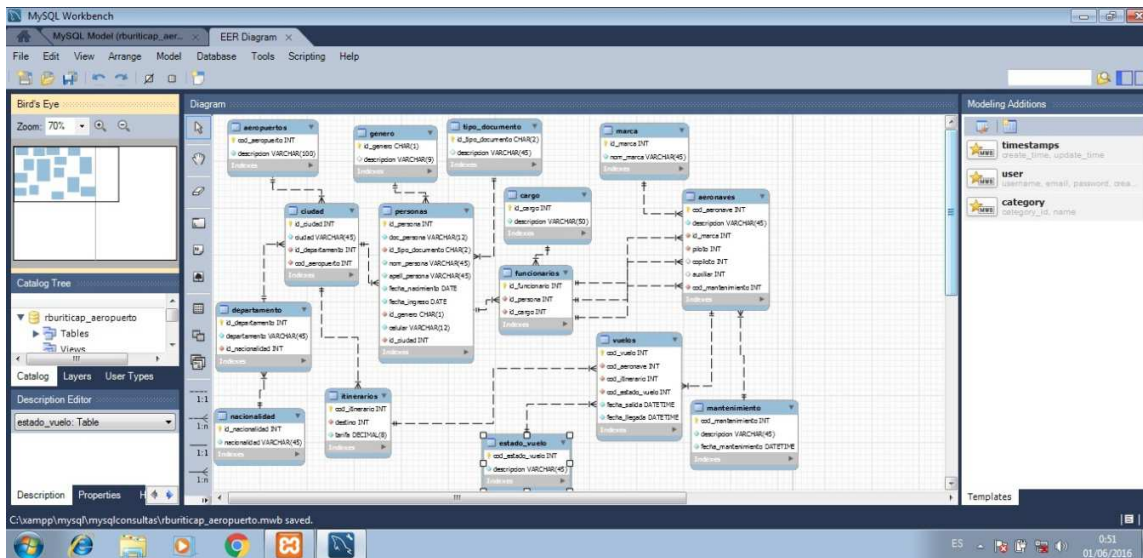
- Comenzamos a crear las tablas



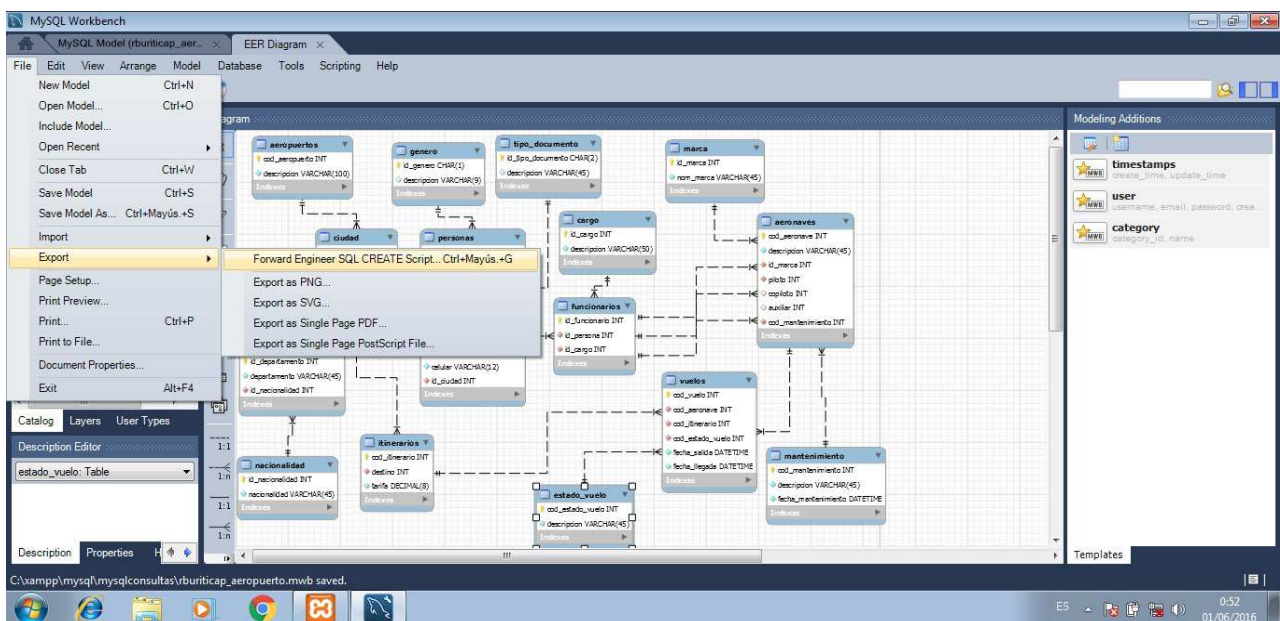
- Creamos las Llaves Foráneas para que se creen las relaciones



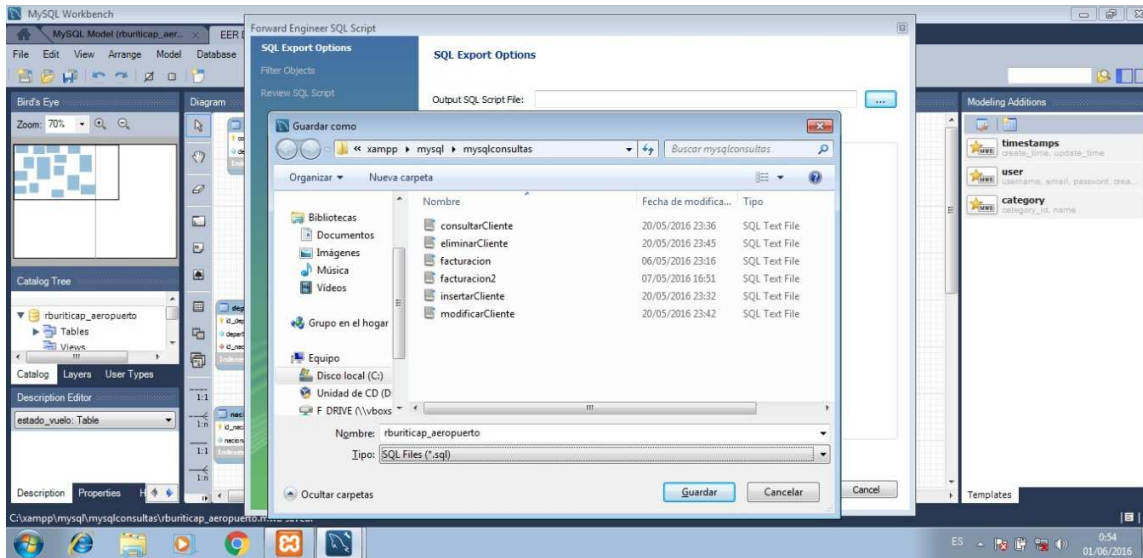
- **Modelo terminado**



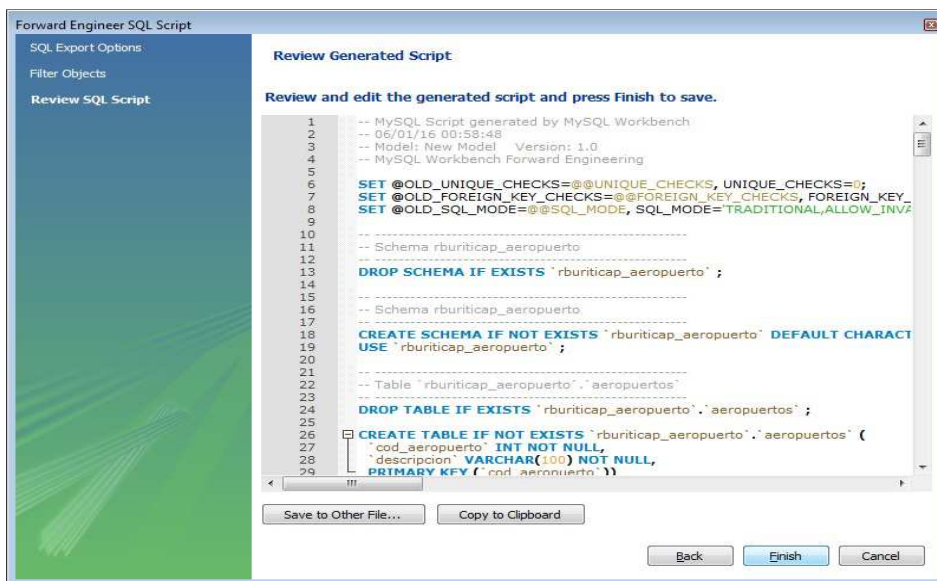
- **Vamos a Exportar el Script de SQL**



- Creamos el archivo para el Script

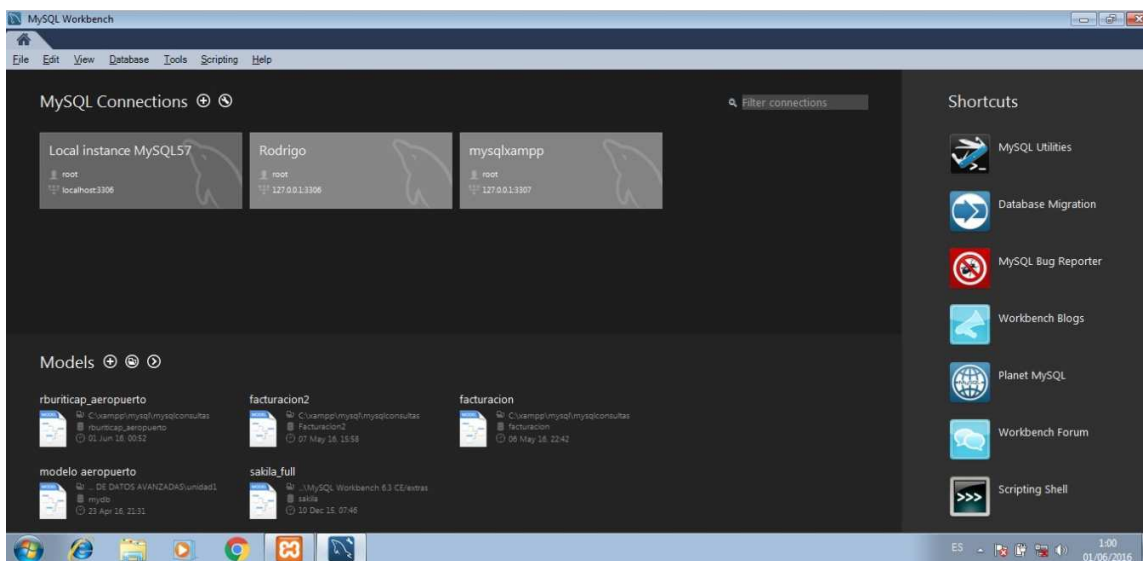


- Vista previa del Script antes de exportarse

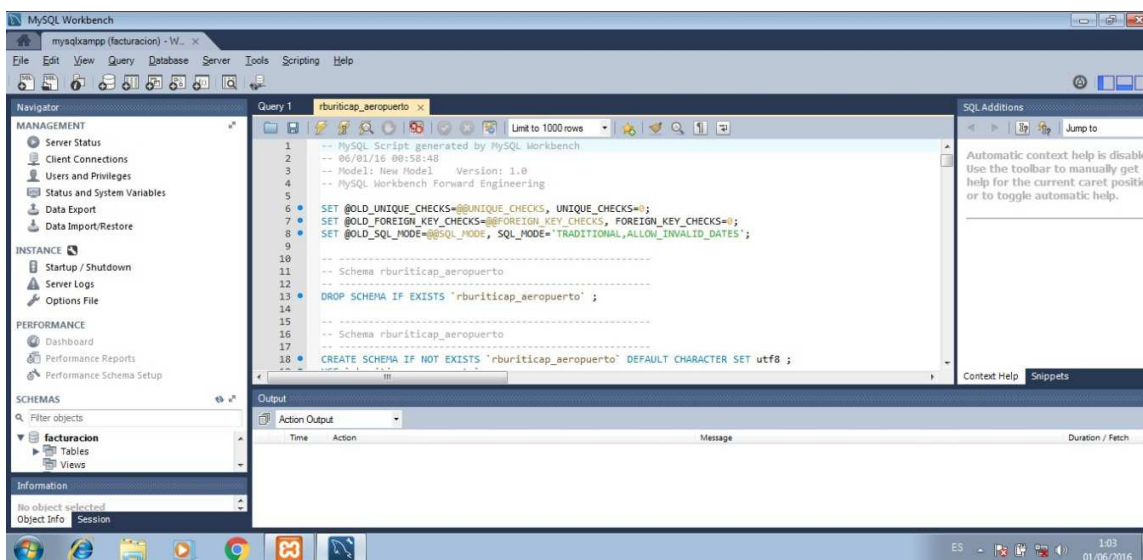


- Entramos a nuestra conexión mysqlxampp creada previamente hacia el servidor mysql del xampp

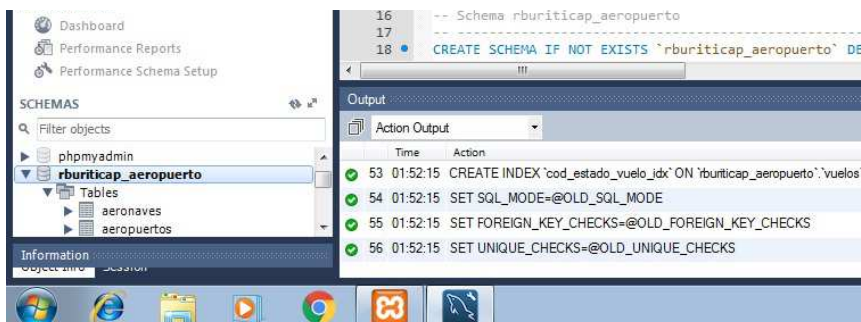




- Abrimos el Script y lo ejecutamos



- En Schemas verificamos que la BD esté creada



## INSERCIÓN DE VALORES EN LA BASE DE DATOS

Para esto se utilizará la sentencia INSERT - INTO - VALUES, línea por línea para agregar los registros a las tablas

- **INSERCIÓN DE DATOS EN LA TABLA NACIONALIDADES**

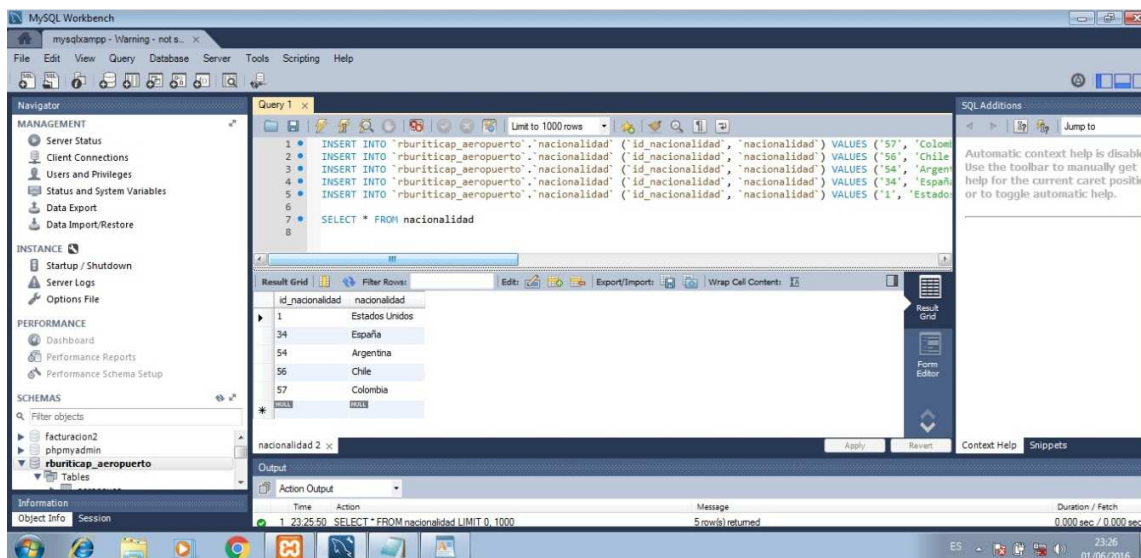
```
INSERT INTO `rburiticap_aeropuerto`.`nacionalidad` (`id_nacionalidad`, `nacionalidad`)
VALUES ('57', 'Colombia');
```

```
INSERT INTO `rburiticap_aeropuerto`.`nacionalidad` (`id_nacionalidad`, `nacionalidad`)
VALUES ('56', 'Chile');
```

```
INSERT INTO `rburiticap_aeropuerto`.`nacionalidad` (`id_nacionalidad`, `nacionalidad`)
VALUES ('54', 'Argentina');
```

```
INSERT INTO `rburiticap_aeropuerto`.`nacionalidad` (`id_nacionalidad`, `nacionalidad`)
VALUES ('34', 'España');
```

```
INSERT INTO `rburiticap_aeropuerto`.`nacionalidad` (`id_nacionalidad`, `nacionalidad`)
VALUES ('1', 'Estados Unidos');
```



- **INSERCIÓN DE DATOS EN LA TABLA DEPARTAMENTOS**

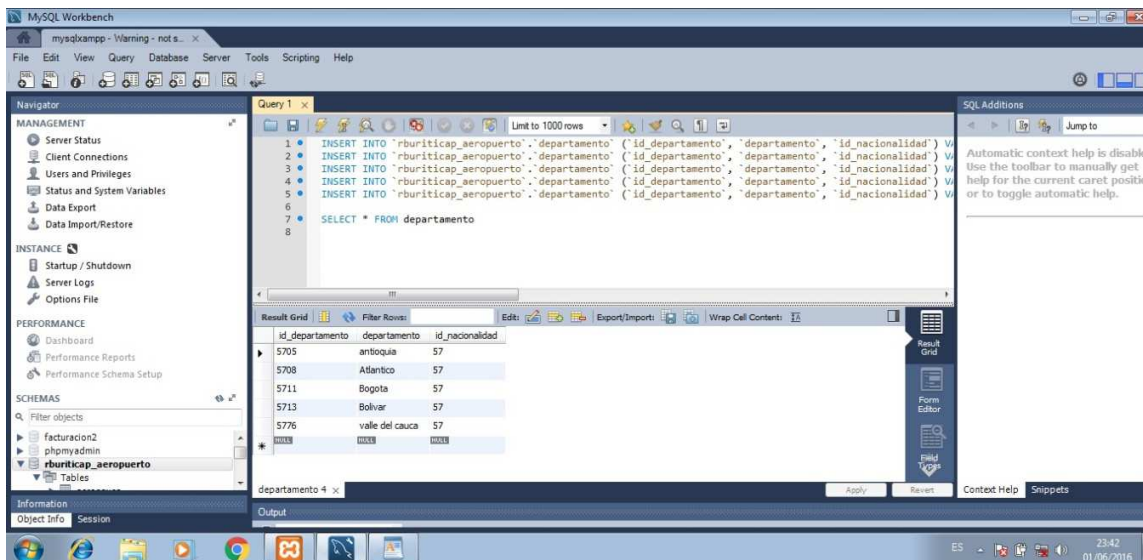
```
INSERT INTO `rburiticap_aeropuerto`.`departamento` (`id_departamento`, `departamento`, `id_nacionalidad`) VALUES ('5705', 'antioquia', '57');
```

```
INSERT INTO `rburiticap_aeropuerto`.`departamento` (`id_departamento`, `departamento`, `id_nacionalidad`) VALUES ('5776', 'valle del cauca', '57');
```

```
INSERT INTO `rburiticap_aeropuerto`.`departamento` (`id_departamento`, `departamento`, `id_nacionalidad`) VALUES ('5711', 'Bogota', '57');
```

```
INSERT INTO `rburiticap_aeropuerto`.`departamento` (`id_departamento`, `departamento`, `id_nacionalidad`) VALUES ('5708', 'Atlantico', '57');
```

```
INSERT INTO `rburiticap_aeropuerto`.`departamento` (`id_departamento`, `departamento`, `id_nacionalidad`) VALUES ('5713', 'Bolívar', '57');
```



- **INSERCIÓN DE DATOS EN LA TABLA AEROPUERTOS**

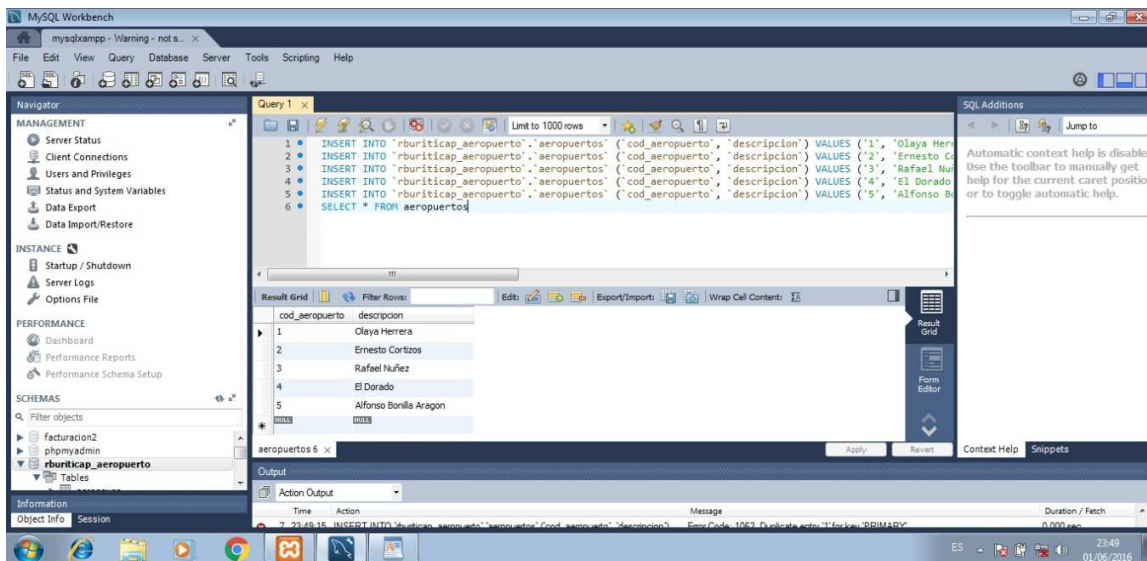
```
INSERT INTO `rburiticap_aeropuerto`.`aeropuertos` (`cod_aeropuerto`, `descripcion`) VALUES ('1', 'Olaya Herrera');
```

```
INSERT INTO `rburiticap_aeropuerto`.`aeropuertos` (`cod_aeropuerto`, `descripcion`) VALUES ('2', 'Ernesto Cortizos');
```

```
INSERT INTO `rburiticap_aeropuerto`.`aeropuertos` (`cod_aeropuerto`, `descripcion`) VALUES ('3', 'Rafael Nuñez');
```

```
INSERT INTO `rburiticap_aeropuerto`.`aeropuertos` (`cod_aeropuerto`, `descripcion`) VALUES ('4', 'El Dorado');
```

```
INSERT INTO `rburiticap_aeropuerto`.`aeropuertos` (`cod_aeropuerto`, `descripcion`)
VALUES ('5', 'Alfonso Bonilla Aragon');
```



## • INSERCIÓN DE DATOS EN LA TABLA CIUDADES

```
INSERT INTO `rburiticap_aeropuerto`.`ciudad` (`id_ciudad`, `ciudad`, `id_departamento`, `cod_aeropuerto`) VALUES ('7601', 'Cali', '5776', '5');
```

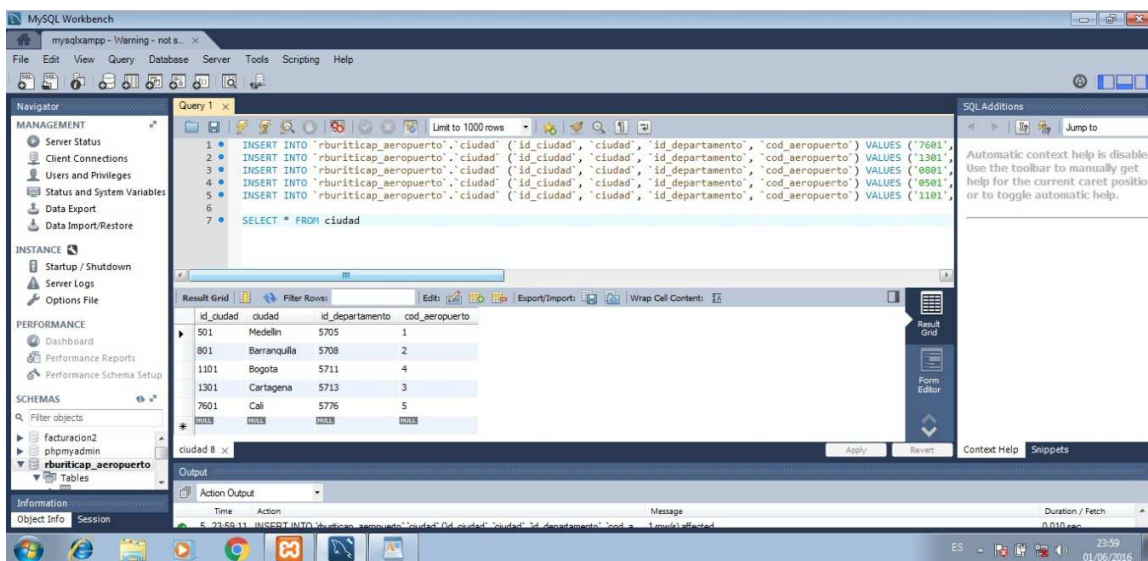
```
INSERT INTO `rburiticap_aeropuerto`.`ciudad` (`id_ciudad`, `ciudad`, `id_departamento`, `cod_aeropuerto`) VALUES ('1301', 'Cartagena', '5713', '3');
```

```
INSERT INTO `rburiticap_aeropuerto`.`ciudad` (`id_ciudad`, `ciudad`, `id_departamento`, `cod_aeropuerto`) VALUES ('0801', 'Barranquilla', '5708', '2');
```

```
INSERT INTO `rburiticap_aeropuerto`.`ciudad` (`id_ciudad`, `ciudad`, `id_departamento`, `cod_aeropuerto`) VALUES ('0501', 'Medellin', '5705', '1');
```

```
INSERT INTO `rburiticap_aeropuerto`.`ciudad` (`id_ciudad`, `ciudad`, `id_departamento`, `cod_aeropuerto`) VALUES ('1101', 'Bogota', '5711', '4');
```





## • INSERCIÓN DE DATOS EN LA TABLA ITINERARIOS

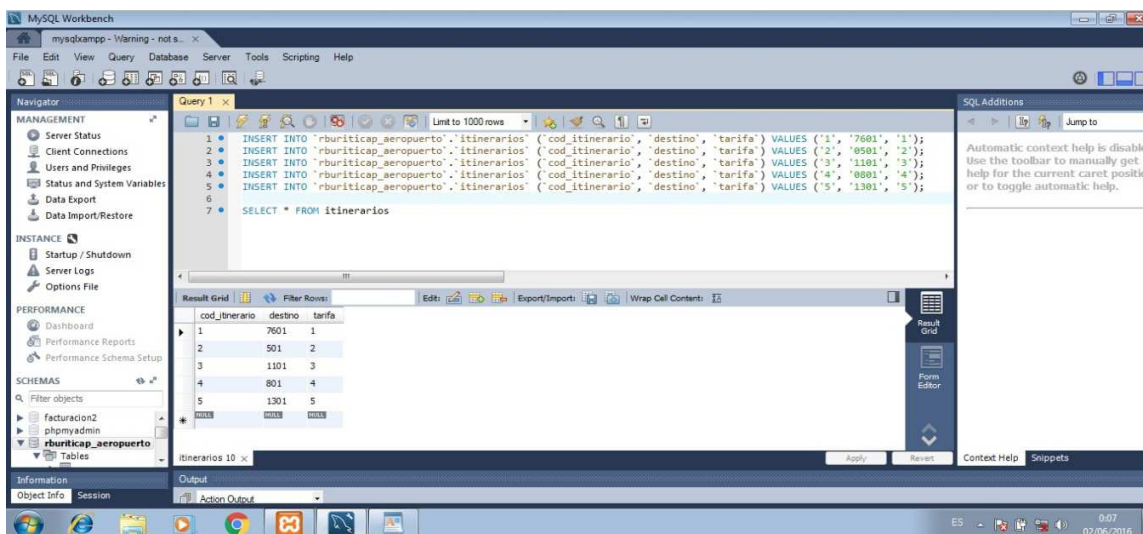
```
INSERT INTO `rburiticap_aeropuerto`.`itinerarios` (`cod_itinerario`, `destino`, `tarifa`)
VALUES ('1', '7601', '1');
```

```
INSERT INTO `rburiticap_aeropuerto`.`itinerarios` (`cod_itinerario`, `destino`, `tarifa`)
VALUES ('2', '0501', '2');
```

```
INSERT INTO `rburiticap_aeropuerto`.`itinerarios` (`cod_itinerario`, `destino`, `tarifa`)
VALUES ('3', '1101', '3');
```

```
INSERT INTO `rburiticap_aeropuerto`.`itinerarios` (`cod_itinerario`, `destino`, `tarifa`)
VALUES ('4', '0801', '4');
```

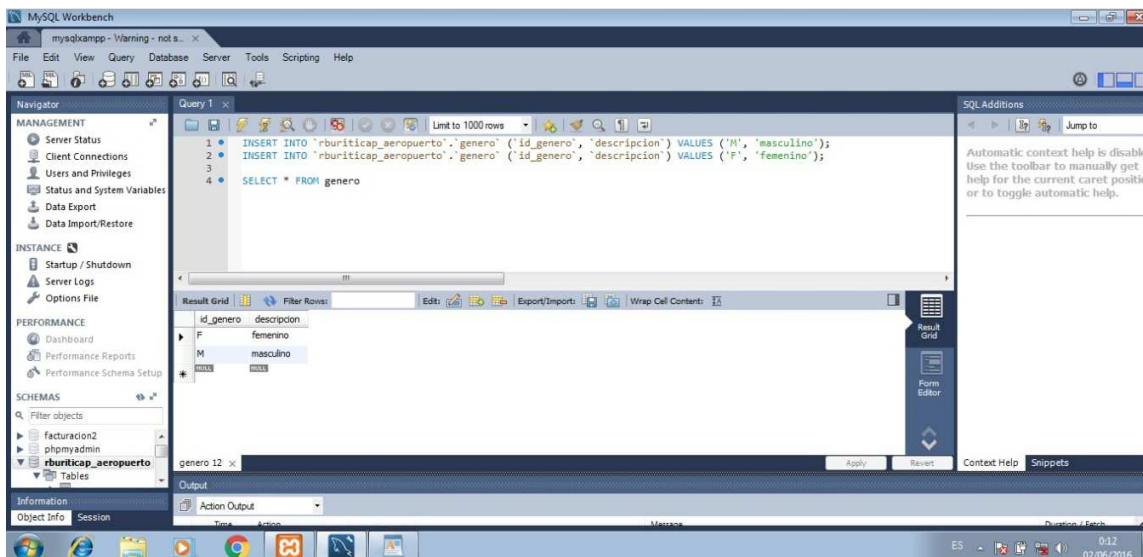
```
INSERT INTO `rburiticap_aeropuerto`.`itinerarios` (`cod_itinerario`, `destino`, `tarifa`)
VALUES ('5', '1301', '5');
```



## • INSERCIÓN DE DATOS EN LA TABLA GENERO

INSERT INTO `rburiticap\_aeropuerto`.`genero` (`id\_genero`, `descripcion`) VALUES ('M', 'masculino');

INSERT INTO `rburiticap\_aeropuerto`.`genero` (`id\_genero`, `descripcion`) VALUES ('F', 'femenino');

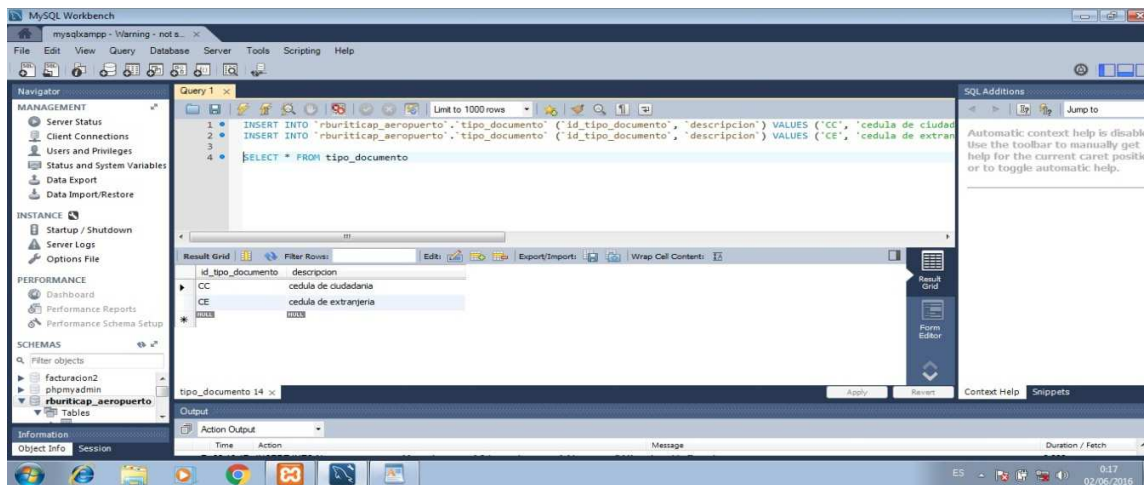


## • INSERCIÓN DE DATOS EN LA TABLA TIPO\_DOCUMENTO

INSERT INTO `rburiticap\_aeropuerto`.`tipo\_documento` (`id\_tipo\_documento`, `descripcion`) VALUES ('CC', 'cedula de ciudadania');

INSERT INTO `rburiticap\_aeropuerto`.`tipo\_documento` (`id\_tipo\_documento`, `descripcion`)

VALUES ('CE', 'cedula de extranjeria');



## • INSERCIÓN DE DATOS EN LA TABLA PERSONAS

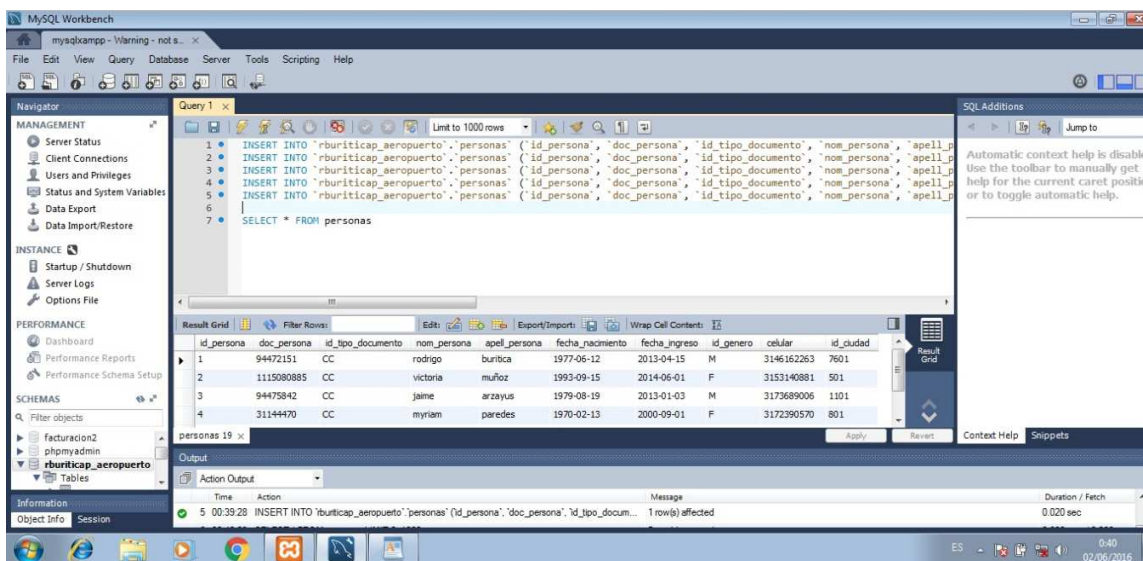
```
INSERT INTO `rburiticap_aeropuerto`.`personas` (`id_persona`, `doc_persona`, `id_tipo_documento`, `nom_persona`, `apell_persona`, `fecha_nacimiento`, `fecha_ingreso`, `id_genero`, `celular`, `id_ciudad`) VALUES ('1', '94472151', 'CC', 'rodrigo', 'buritica', '1977-06-12', '2013-04-15', 'M', '3146162263', '7601');
```

```
INSERT INTO `rburiticap_aeropuerto`.`personas` (`id_persona`, `doc_persona`, `id_tipo_documento`, `nom_persona`, `apell_persona`, `fecha_nacimiento`, `fecha_ingreso`, `id_genero`, `celular`, `id_ciudad`) VALUES ('2', '1115080885', 'CC', 'victoria', 'muñoz', '1993-09-15', '2014-06-01', 'F', '3153140881', '0501');
```

```
INSERT INTO `rburiticap_aeropuerto`.`personas` (`id_persona`, `doc_persona`, `id_tipo_documento`, `nom_persona`, `apell_persona`, `fecha_nacimiento`, `fecha_ingreso`, `id_genero`, `celular`, `id_ciudad`) VALUES ('3', '94475842', 'CC', 'jaime', 'arzayus', '1979-08-19', '2013-01-03', 'M', '3173689006', '1101');
```

```
INSERT INTO `rburiticap_aeropuerto`.`personas` (`id_persona`, `doc_persona`, `id_tipo_documento`, `nom_persona`, `apell_persona`, `fecha_nacimiento`, `fecha_ingreso`, `id_genero`, `celular`, `id_ciudad`) VALUES ('4', '31144470', 'CC', 'myriam', 'paredes', '1970-02-13', '2000-09-01', 'F', '3172390570', '0801');
```

```
INSERT INTO `rburiticap_aeropuerto`.`personas` (`id_persona`, `doc_persona`, `id_tipo_documento`, `nom_persona`, `apell_persona`, `fecha_nacimiento`, `fecha_ingreso`, `id_genero`, `celular`, `id_ciudad`) VALUES ('5', '14899718', 'CC', 'jairo', 'dominguez', '1976-03-20', '2014-08-01', 'M', '3177842678', '1301');
```

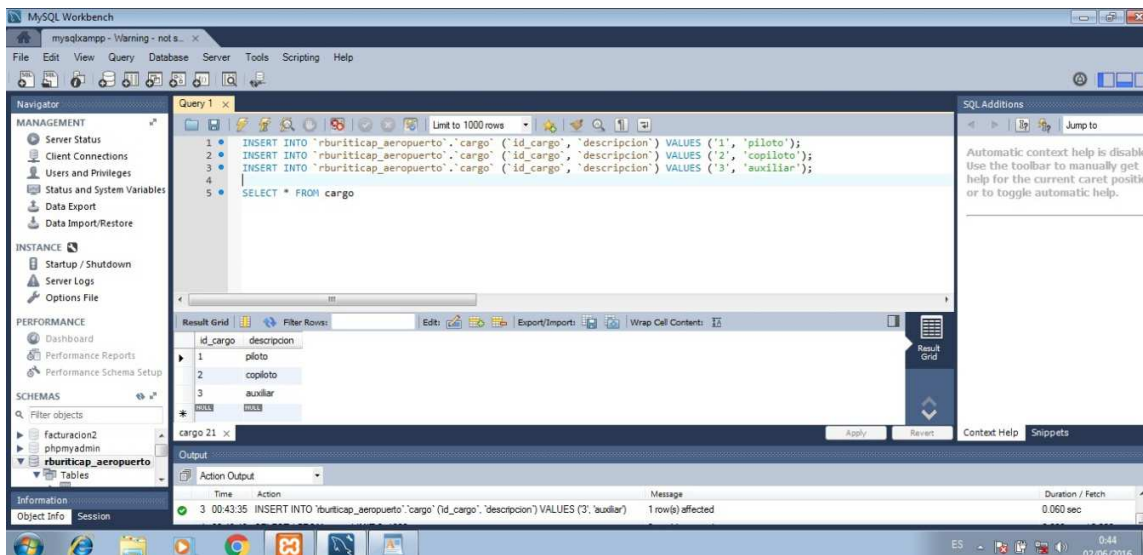


## • INSERCIÓN DE DATOS EN LA TABLA CARGO

```
INSERT INTO `rburiticap_aeropuerto`.`cargo` (`id_cargo`, `descripcion`) VALUES ('1', 'piloto');
```

```
INSERT INTO `rburiticap_aeropuerto`.`cargo` (`id_cargo`, `descripcion`) VALUES ('2', 'copiloto');
```

```
INSERT INTO `rburiticap_aeropuerto`.`cargo` (`id_cargo`, `descripcion`) VALUES ('3', 'auxiliar');
```



## • INSERCIÓN DE DATOS EN LA TABLA FUNCIONARIOS

```
INSERT INTO `rburiticap_aeropuerto`.`funcionarios` (`id_funcionario`, `id_persona`,
```

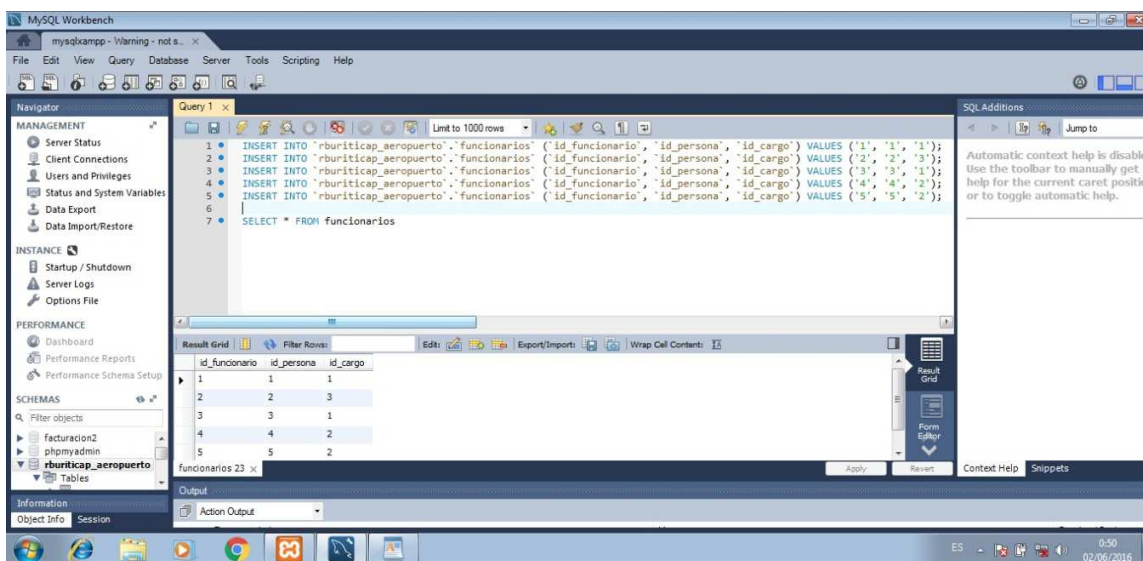
```
`id_cargo`) VALUES ('1', '1', '1');
```

```
INSERT INTO `rburiticap_aeropuerto`.`funcionarios` (`id_funcionario`, `id_persona`,  
`id_cargo`) VALUES ('2', '2', '3');
```

```
INSERT INTO `rburiticap_aeropuerto`.`funcionarios` (`id_funcionario`, `id_persona`,  
`id_cargo`) VALUES ('3', '3', '1');
```

```
INSERT INTO `rburiticap_aeropuerto`.`funcionarios` (`id_funcionario`, `id_persona`,  
`id_cargo`) VALUES ('4', '4', '2');
```

```
INSERT INTO `rburiticap_aeropuerto`.`funcionarios` (`id_funcionario`, `id_persona`,  
`id_cargo`) VALUES ('4', '5', '2');
```



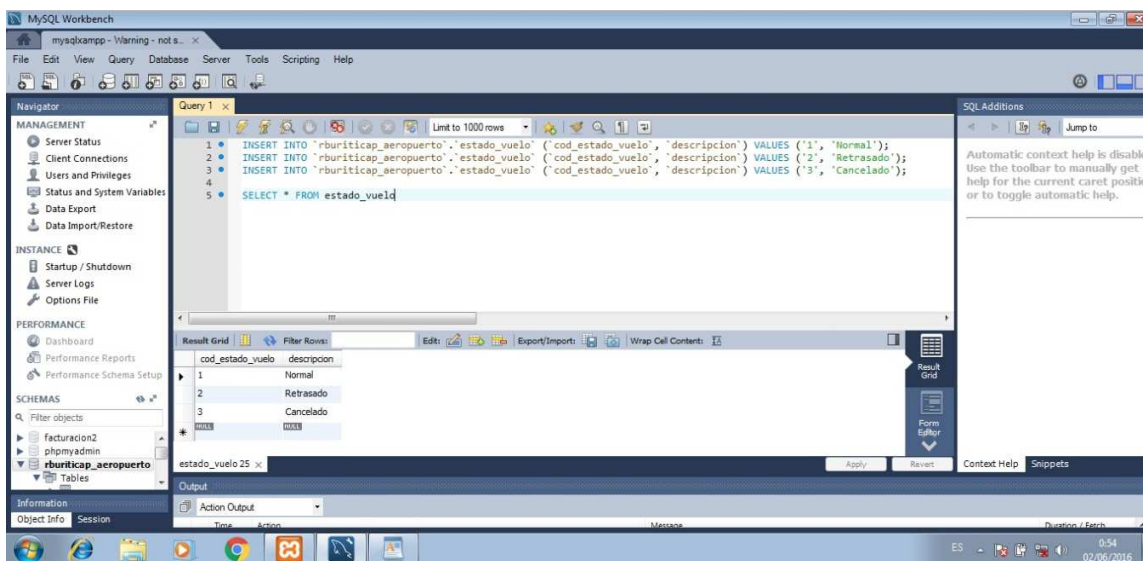
- **INSERCIÓN DE DATOS EN LA TABLA ESTADO\_VUELO**

```
INSERT INTO `rburiticap_aeropuerto`.`estado_vuelo` (`cod_estado_vuelo`, `descripcion`)  
VALUES ('1', 'Normal');
```

```
INSERT INTO `rburiticap_aeropuerto`.`estado_vuelo` (`cod_estado_vuelo`, `descripcion`)  
VALUES ('2', 'Retrasado');
```

```
INSERT INTO `rburiticap_aeropuerto`.`estado_vuelo` (`cod_estado_vuelo`, `descripcion`)  
VALUES ('3', 'Cancelado');
```

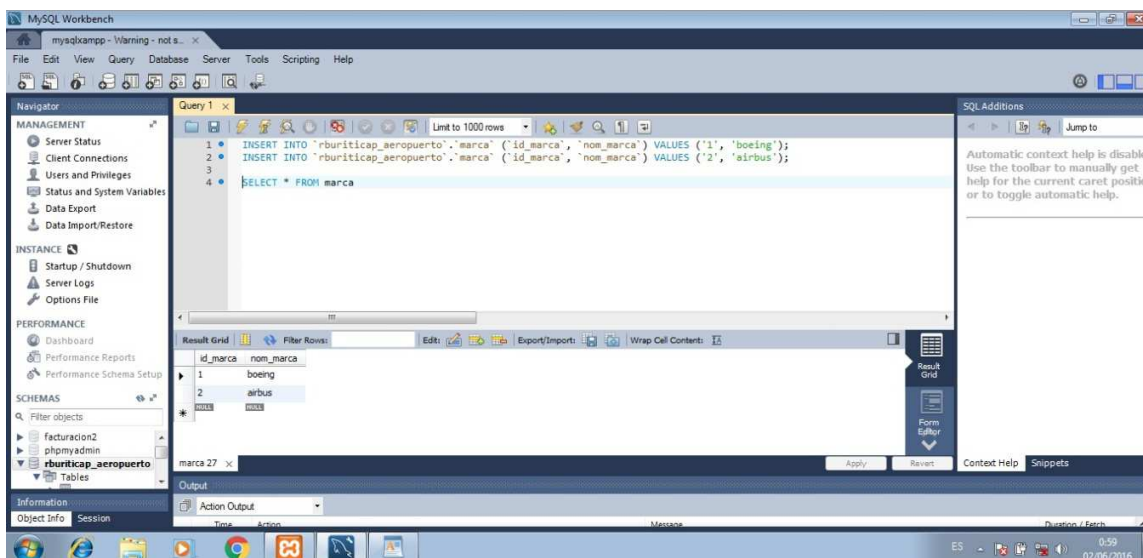




## - **INSERCIÓN DE DATOS EN LA TABLA MARCA**

INSERT INTO `rburiticap\_aeropuerto`.`marca` (`id\_marca`, `nom\_marca`) VALUES ('1', 'boeing');

INSERT INTO `rburiticap\_aeropuerto`.`marca` (`id\_marca`, `nom\_marca`) VALUES ('2', 'airbus');



## - **INSERCIÓN DE DATOS EN LA TABLA MANTENIMIENTO**

INSERT INTO `rburiticap\_aeropuerto`.`mantenimiento` (`cod\_mantenimiento`, `descripcion`, `fecha\_mantenimiento`) VALUES ('1', 'preventivo', '2016-01-15');

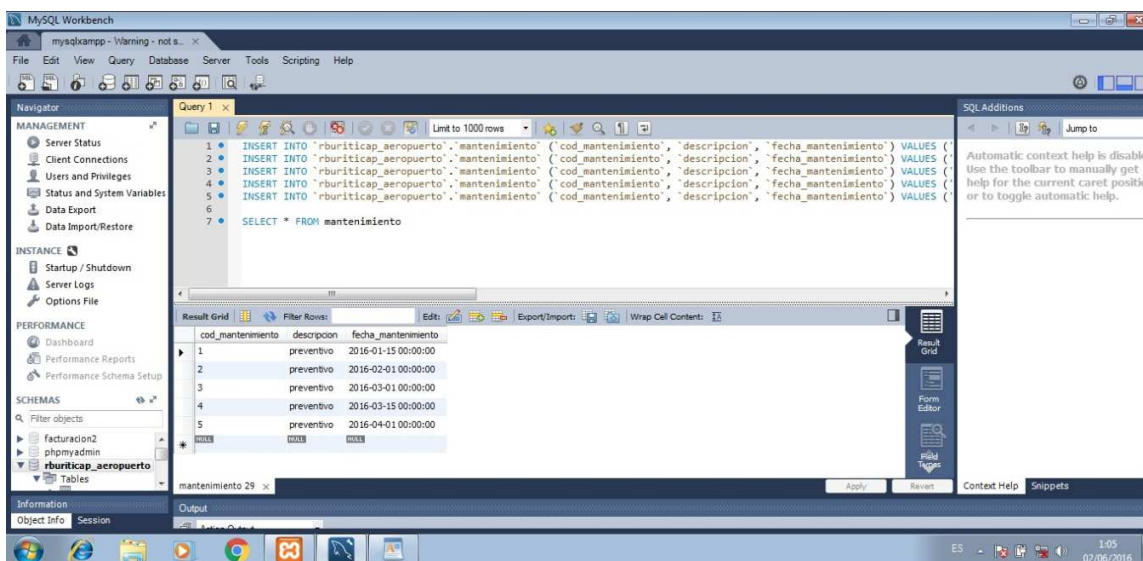
INSERT INTO `rburiticap\_aeropuerto`.`mantenimiento` (`cod\_mantenimiento`, `descripcion`,

```
`fecha_mantenimiento`) VALUES ('2', 'preventivo', '2016-02-01');
```

```
INSERT INTO `rburiticap_aeropuerto`.`mantenimiento` (`cod_mantenimiento`, `descripcion`,  
`fecha_mantenimiento`) VALUES ('3', 'preventivo', '2016-03-01');
```

```
INSERT INTO `rburiticap_aeropuerto`.`mantenimiento` (`cod_mantenimiento`, `descripcion`,  
`fecha_mantenimiento`) VALUES ('4', 'preventivo', '2016-03-15');
```

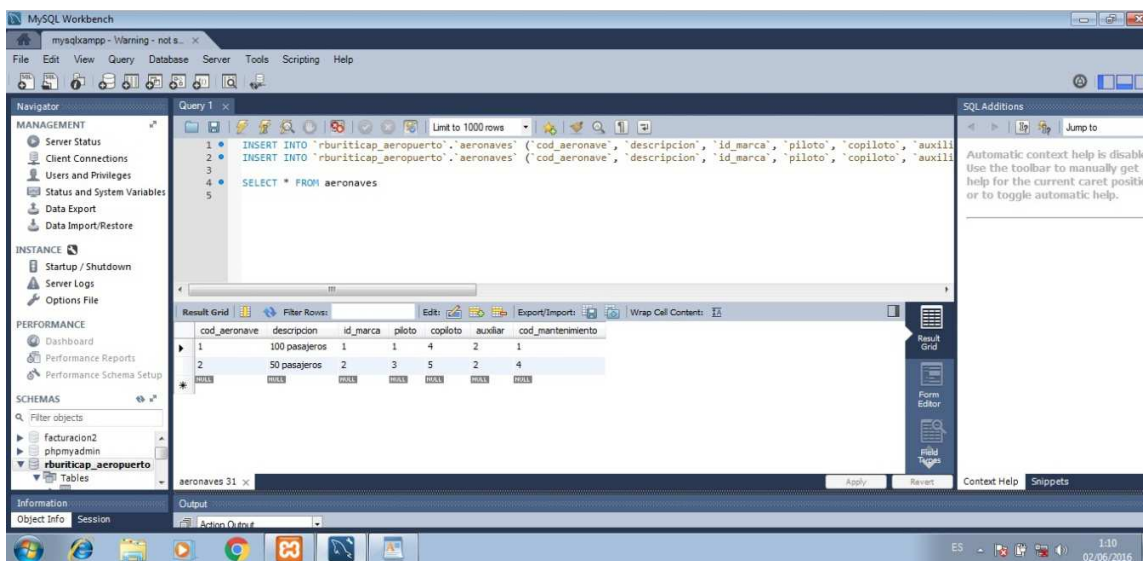
```
INSERT INTO `rburiticap_aeropuerto`.`mantenimiento` (`cod_mantenimiento`, `descripcion`,  
`fecha_mantenimiento`) VALUES ('5', 'preventivo', '2016-04-01');
```



## • INSERCIÓN DE DATOS EN LA TABLA AERONAVES

```
INSERT INTO `rburiticap_aeropuerto`.`aeronaves` (`cod_aeronave`, `descripcion`,  
`id_marca`, `piloto`, `copiloto`, `auxiliar`, `cod_mantenimiento`) VALUES ('1', '100 pasajeros',  
'1', '1', '4', '2', '1');
```

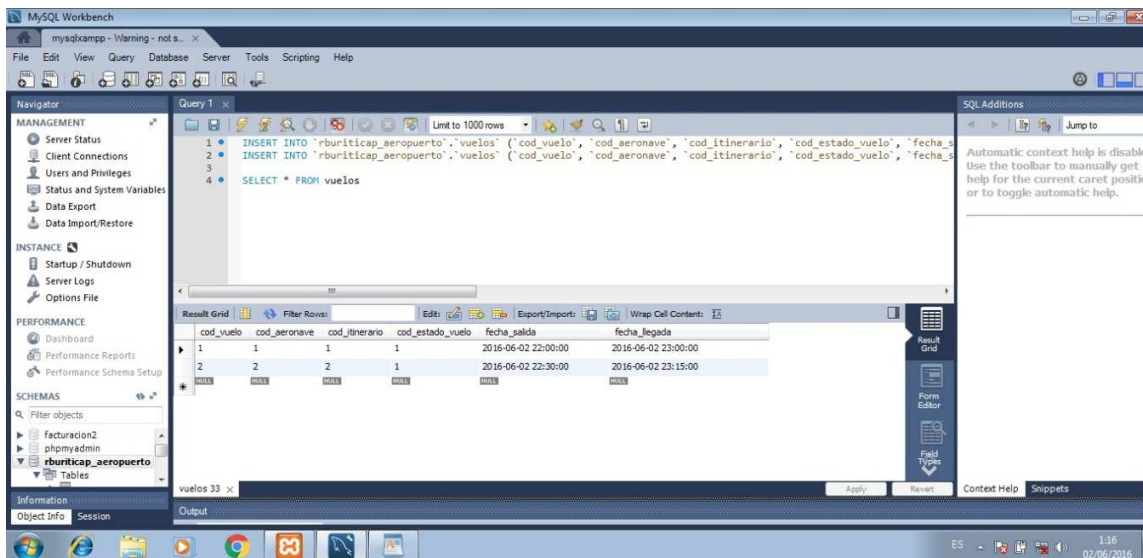
```
INSERT INTO `rburiticap_aeropuerto`.`aeronaves` (`cod_aeronave`, `descripcion`,  
`id_marca`, `piloto`, `copiloto`, `auxiliar`, `cod_mantenimiento`) VALUES ('2', '50 pasajeros',  
'2', '3', '5', '2', '4');
```



## • INSERCIÓN DE DATOS EN LA TABLA VUELOS

INSERT INTO `rburiticap\_aeropuerto`.`vuelos` (`cod\_vuelo`, `cod\_aeronave`, `cod\_itinerario`, `cod\_estado\_vuelo`, `fecha\_salida`, `fecha\_llegada`) VALUES ('1', '1', '1', '1', '2016-06-02 22:00', '2016-06-02 23:00');

INSERT INTO `rburiticap\_aeropuerto`.`vuelos` (`cod\_vuelo`, `cod\_aeronave`, `cod\_itinerario`, `cod\_estado\_vuelo`, `fecha\_salida`, `fecha\_llegada`) VALUES ('2', '2', '2', '1', '2016-06-02 22:30', '2016-06-02 23:15');





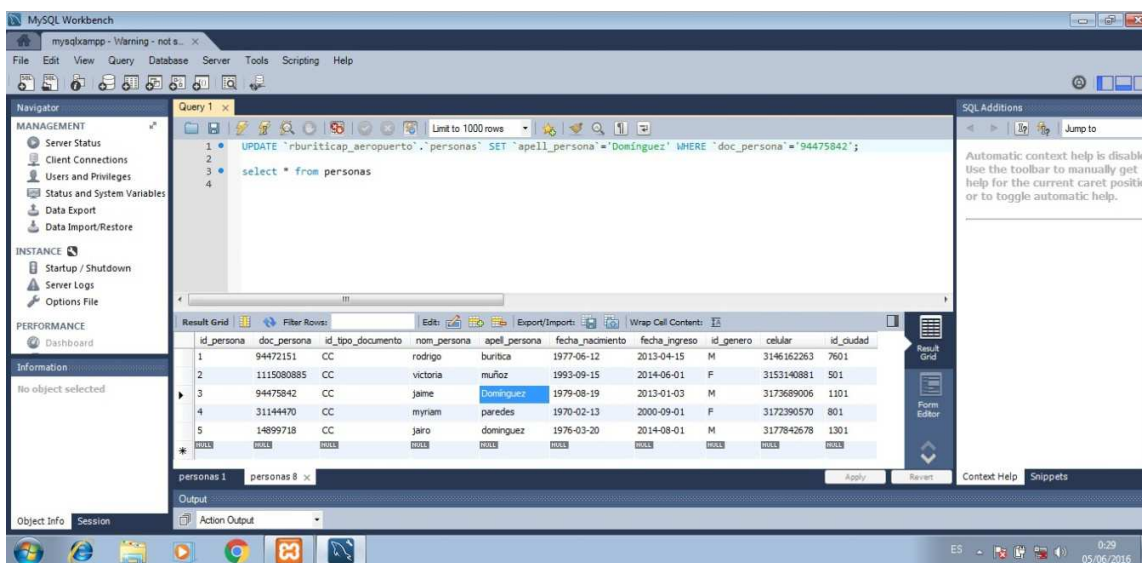
## ACTUALIZACION DE VALORES EN UNA TABLA

Utilizamos la sentencia UPDATE - SET de la siguiente manera:

```
UPDATE nombre_tabla SET columna_a_modificar=nuevo_valor WHERE columna_de_condicion=valor_condicion;
```

Por ejemplo, en nuestro caso en la tabla "personas" tenemos a Jaime Arzayús, el cual tiene por número de cédula 94475842. Cambiaremos su apellido por Dominguez, así:

```
update personas set apell_persona='Dominguez' where doc_persona='94475842';
```



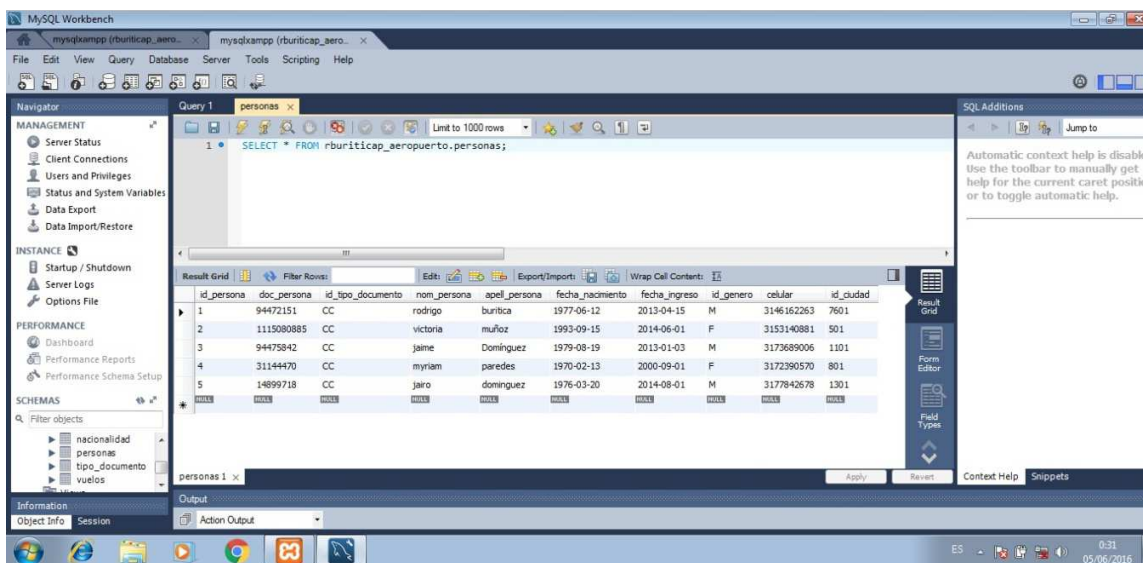
## CONSULTA SENCILLA

Se utilizan las sentencias SELECT - FROM -WHERE de la siguiente manera:

```
SELECT columnas_a_mostrar FROM nombre_tabla WHERE condicion;
```

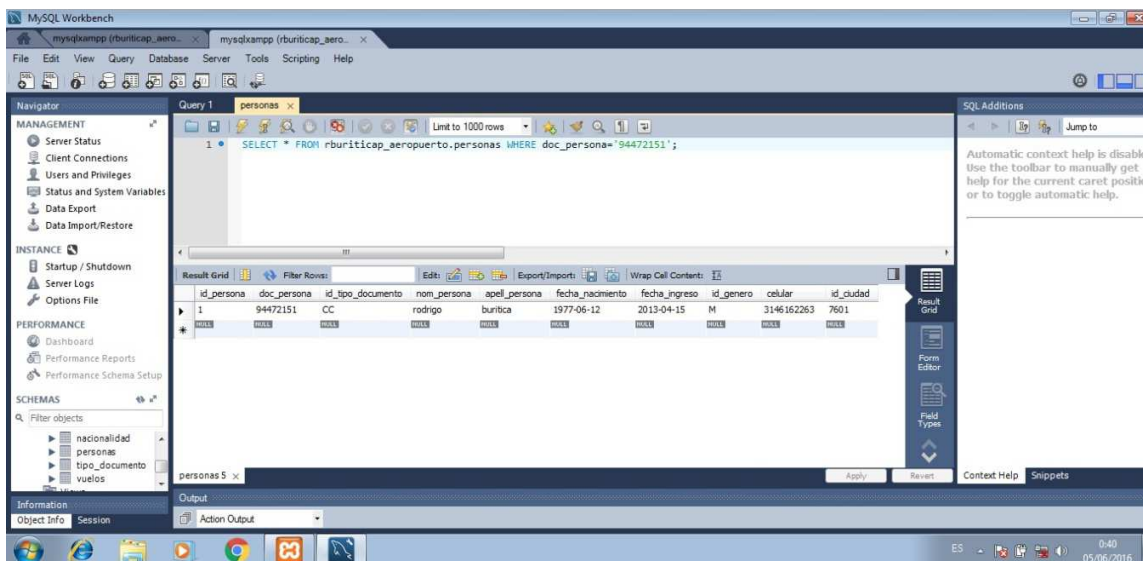
Ejemplo 1: Mostrar todos los datos de la tabla personas.

```
SELECT * FROM personas;
```



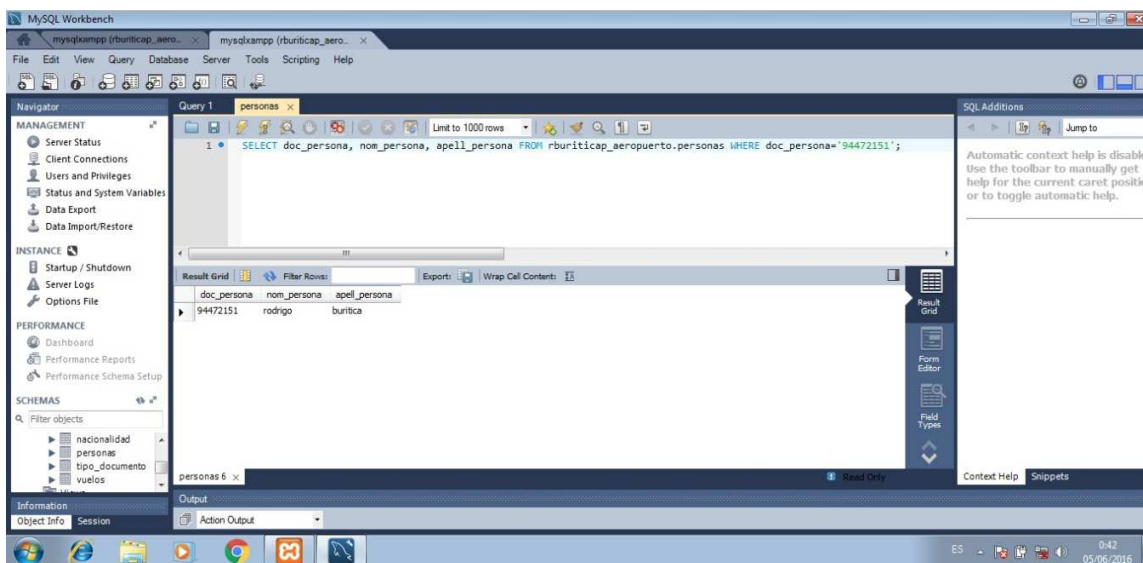
Ejemplo 2: Mostrar completo el registro de la persona cuya cedula es 94472151.

`SELECT * FROM personas WHERE doc_persona='94472151';`



Ejemplo 3: Mostrar solo la cedula, el nombre y el apellido de la persona cuya cedula es 94472151.

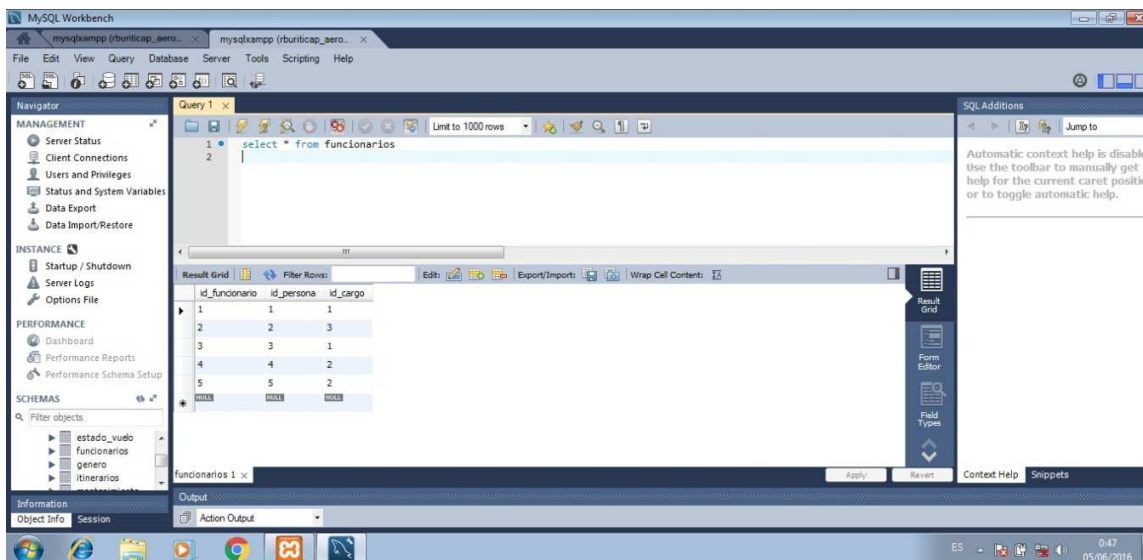
`SELECT doc_persona, nom_persona, apel_persona FROM personas WHERE doc_persona='94472151';`



## CONSULTAS COMPUESTAS

En este caso estamos llamando "consultas compuestas" a aquellas en que involucramos datos de dos o más tablas. Lo que es muy útil en casos como el siguiente, por ejemplo:

Si consultamos sencillamente la tabla funcionarios, tendremos algo así:

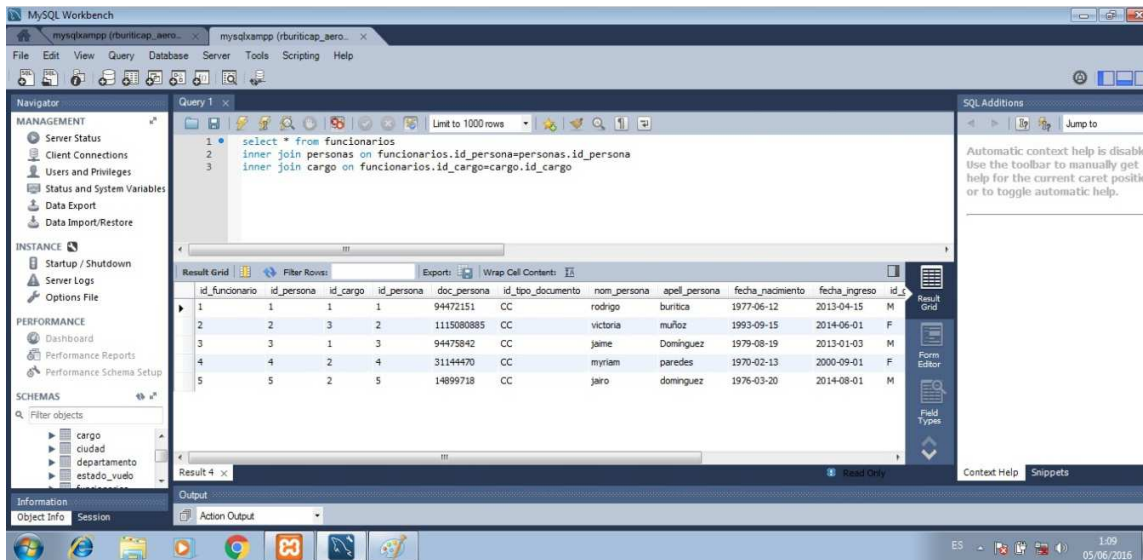


Entonces quisiéramos ver esta información, teniendo también el nombre y el apellido de esas personas y el nombre de esos cargos. Entonces utilizamos la sentencia INNER JOIN para agregar la tabla de personas y la tabla de cargo. La forma de utilizar esta sentencia es:

`INNER JOIN tabla_a_agregar ON tabla_principal_consulta.columna_relacionada = tabla_a_agregar.columna_relacionada`

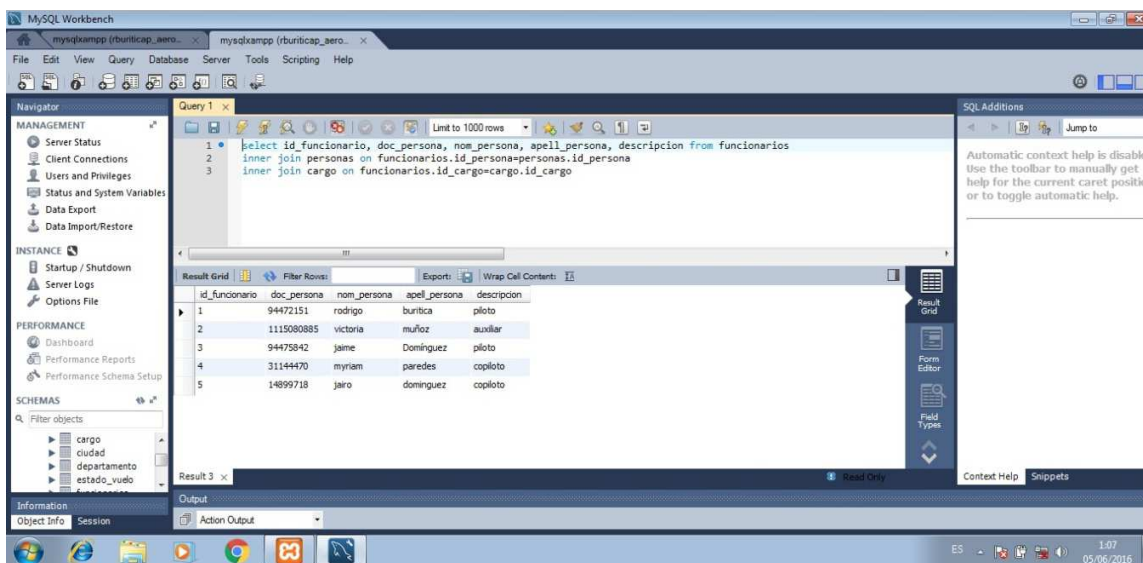
Tendríamos algo así:

```
SELECT * FROM funcionarios  
INNER JOIN personas ON funcionarios.id_persona=personas.id_persona  
INNER JOIN cargo ON funcionarios.id_cargo=cargo.id_cargo
```



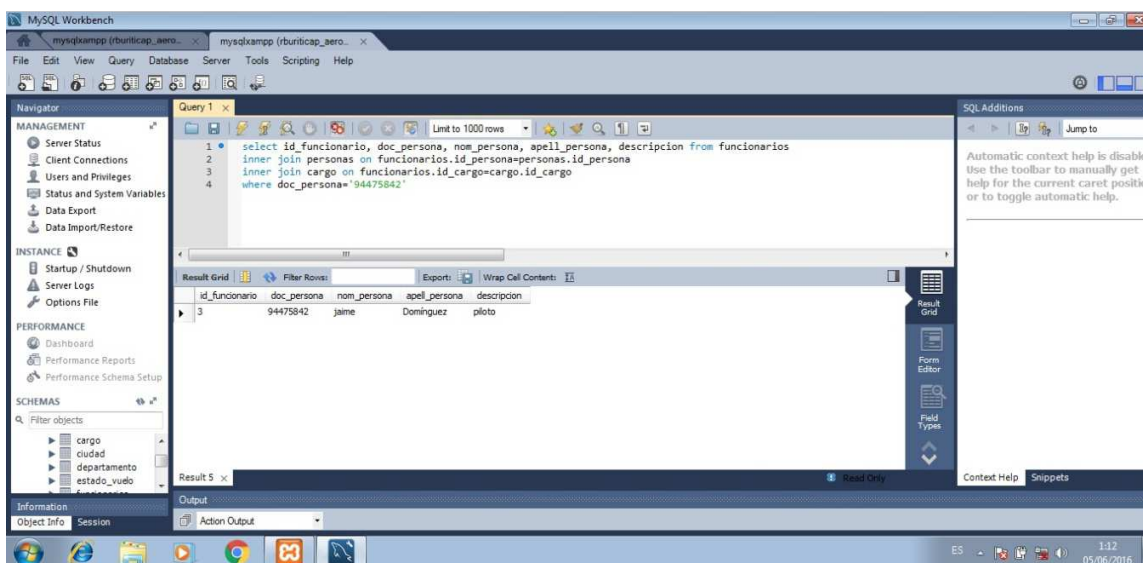
Lo que vemos es que nos está mostrando entonces todas las columnas de las 3 tablas para los datos que tienen relación entre ellas, pero nosotros solo queremos ver el id del funcionario, su cedula, su nombre, su apellido y el nombre de su cargo. Entonces haremos así:

```
SELECT id_funcionario, doc_persona, nom_persona, apell_persona, descripcion FROM  
funcionarios  
INNER JOIN personas ON funcionarios.id_persona=personas.id_persona  
INNER JOIN cargo ON funcionarios.id_cargo=cargo.id_cargo
```



También podemos agregar el WHERE si por ejemplo solo queremos consultar esta información para el funcionario cuya cedula sea 94475842, así:

```
SELECT id_funcionario, doc_persona, nom_persona, apell_persona, descripcion FROM
funcionarios
INNER JOIN personas ON funcionarios.id_persona=personas.id_persona
INNER JOIN cargo ON funcionarios.id_cargo=cargo.id_cargo
WHERE doc_persona='94475842'
```



## ELIMINACIÓN DE REGISTROS

Se utiliza la sentencia DELETE FROM - WHERE de la siguiente manera:

DELETE FROM tabla WHERE columna\_condicion=valor\_condicion

Por ejemplo, si queremos eliminar el segundo vuelo de la tabla vuelos, haríamos:

DELETE FROM vuelos WHERE cod\_vuelo='2'



MySQL Workbench

mysqlxamp (rburiticap\_aero...) mysqlxamp (rburiticap\_aero...)

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

- nacionalidad
- personas
- tipo\_documento
- vuelos

Query 1

```
1 select * from vuelos
```

Result Grid

cod_vuelo	cod_aerolinea	cod_itinerario	cod_estado_vuelo	fecha_salida	fecha_llegada
1	1	1	1	2016-06-02 22:00:00	2016-06-02 23:00:00
2	2	2	1	2016-06-02 22:30:00	2016-06-02 23:15:00

vuelos 6 x

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

ES 1:17 05/06/2016

MySQL Workbench

mysqlxamp (rburiticap\_aero...) mysqlxamp (rburiticap\_aero...)

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

- nacionalidad
- personas
- tipo\_documento
- vuelos

Query 1

```
1 DELETE FROM "rburiticap_aeropuerto"."vuelos" WHERE "cod_vuelo" = '2';
2
3 SELECT * FROM vuelos
4
```

Result Grid

cod_vuelo	cod_aerolinea	cod_itinerario	cod_estado_vuelo	fecha_salida	fecha_llegada
1	1	1	1	2016-06-02 22:00:00	2016-06-02 23:00:00

vuelos 7 x

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

ES 1:21 05/06/2016

## CONCLUSIONES

- Este lenguaje de MySQL es muy, muy similar al SQL server, ya que los comandos utilizados por este motor son iguales a los comando utilizados en el motor de SQL, por lo cual se hace muy amigable para un desarrollador que quiera migrar o comenzar a usar este motor de base de datos.
- El desarrollo de esta práctica nos permite recordar, implementar y utilizar estos comandos, ya que con el tiempo nos acostumbramos al uso del administrador de consultas o realizar los cambios directamente en las tablas que cuando vamos a presentar una entrevista o no contamos con esa herramienta nos es difícil la transacción de consultas con la base de datos.



## **BIBLIOGRAFIA**

- Martín Escofet, Carme. El lenguaje SQL. P06/M2109/02149
- García, Alvaro E. Manual práctico de SQL orientado a SQL 7.0
- Oracle and/or affiliates. MySQL 5.0 Reference Manual. 1997-2011