

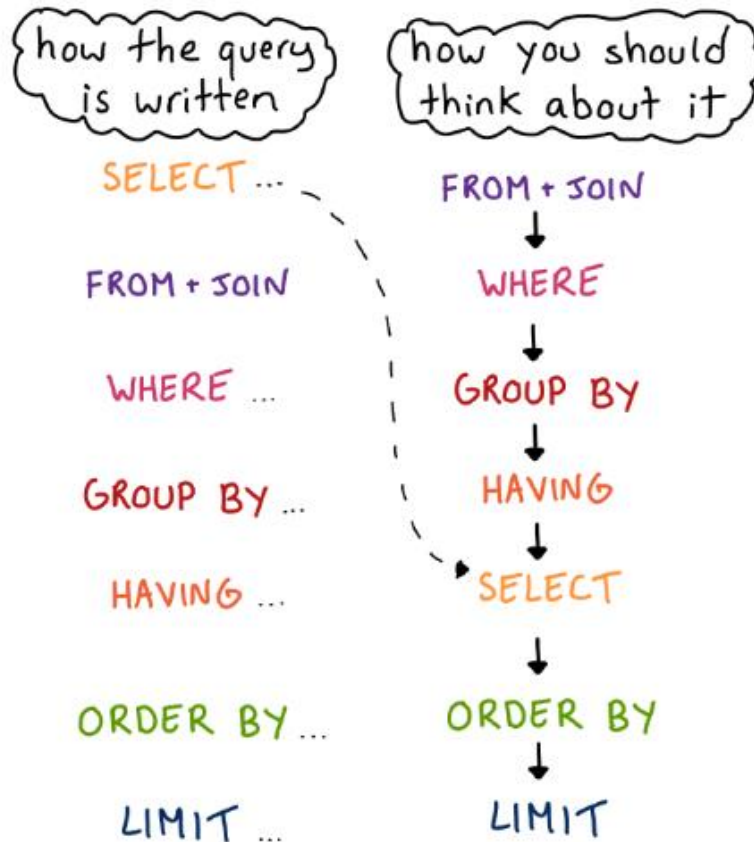
SQL

Session 4



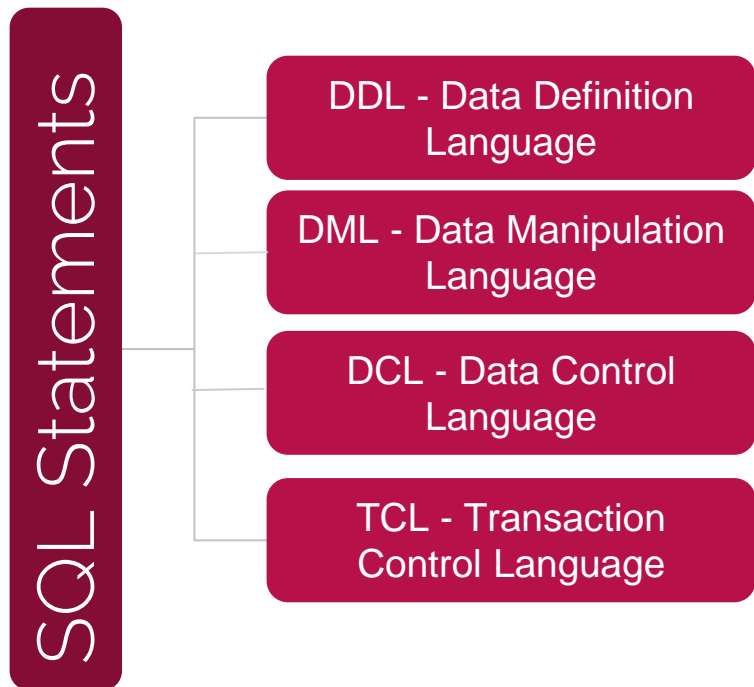


The query's steps don't happen in the order they're written:



(In reality query execution is much more complicated than this.
There are a lot of optimizations.)

Introduction





DDL Commands

Table of Contents



- ▶ Introduction
- ▶ Data Types
- ▶ CREATE TABLE
- ▶ ALTER TABLE
- ▶ DROP TABLE



▶ Data Definition Language

- DDL specifies the database schema.
- Some statements used in DDL are **CREATE, ALTER, DROP.**
- DDL statements are typically used to set up and configure a new database before we insert data.



Data Manipulation Language



- Data Manipulation Language (DML) enables users to access or manipulate data.
- **INSERT, UPDATE, DELETE, SELECT*** are the statements used in DML.

* In some sources, SELECT statement is grouped into a different category called DQL (Data Query Language).



▶ Data Control Language

- Data Control Language (DCL) is used to grant or revoke access control.
- Its statements are **REVOKE** and **GRANT**.



Transaction Control Language



- Transaction Control Language (TCL) controls the transactions of DML and DDL commands.
- Some statements in TCL are **COMMIT, ROLLBACK, SAVEPOINT.**



2

Data Types

▶ Data Types



The data type of a column defines what value the column can hold: integer, character, date and time, binary, and so on.

▶ Data Types



String

Date and
Time

Numeric

String Data Types



The string data types are:

- CHAR
- VARCHAR
- BINARY
- VARBINARY
- BLOB
- TEXT
- ENUM
- SET



▶ Date and Time Data Types

The date and time data types are:

- DATE
- DATETIME
- TIMESTAMP
- YEAR



Numeric Data Types

Integer Types (Exact Value)

- INTEGER or INT
- SMALLINT
- TINYINT
- MEDIUMINT
- BIGINT

Floating-Point Types (Approximate Value)

- FLOAT
- DOUBLE

Fixed-Point Types (Exact Value)

- DECIMAL
- NUMERIC

▶ Data Types



Data types might have different names in different database. And even if the name is the same, the size and other details may be different! Always check the documentation!



3

CREATE TABLE

CREATE TABLE



When creating a table, we use **CREATE TABLE** statement.

Syntax of a Basic Create Table Statement

```
CREATE TABLE table_name  
    (column_name1 data_type,  
     column_name2 data_type);
```



CREATE TABLE-Example



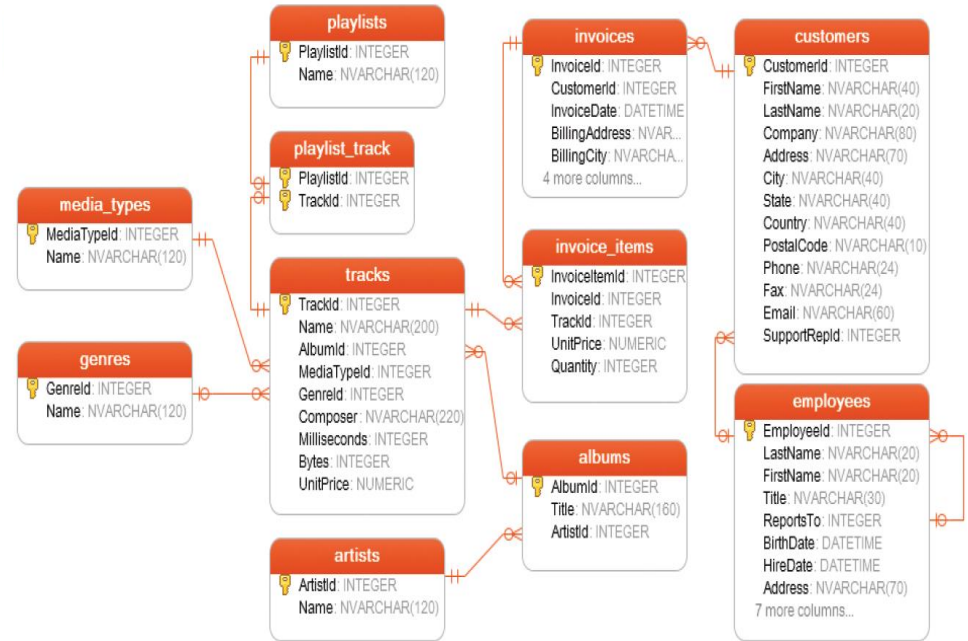
```
CREATE TABLE employee
    (first_name VARCHAR(15),
     last_name  VARCHAR(20),
     age        INT,
     hire_date  DATE);
```

Note: Values in VARCHAR columns are variable-length strings. The length can be specified as a value from 0 to 65,535.

Query Time

Please add a table to your existing chinook database:
The table name will be **leaves** we will use it to keep record of the employees' annual or sick leaves
Column names:

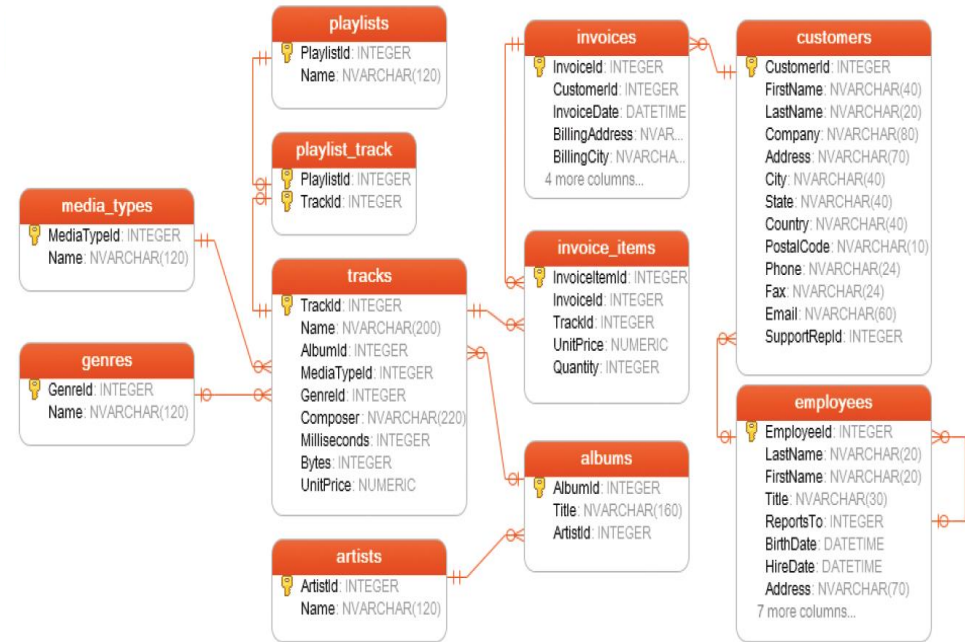
- id
- employee_id
- start_date
- end_date





Please add a table to your existing sql_course database:
The table name will be **vacation_plan** of the employees for this summer.
Column names:

- place_id
- country
- hotel_name
- employee_id
- vacation_length
- budget



▶ DROP TABLE



The DROP TABLE statement is used to drop an existing table in a database.

Syntax:

```
DROP TABLE table_name;
```

```
TRUNCATE TABLE table_name;
```



4

ALTER TABLE

▶ ALTER TABLE



The **ALTER TABLE** statement is used to add, delete, or modify columns in an existing table. It is also used to add and drop various constraints on an existing table.

To add a column in a table, use the following syntax:

```
ALTER TABLE table_name  
ADD column_name data_type;
```




Add a column to your vacation_plan table named "city".

```
ALTER TABLE table_name  
ADD column_name data_type;
```

▶ ALTER TABLE



To delete a column in a table, use the following syntax:

```
ALTER TABLE table_name  
DROP column_name;
```

To change the data type of a column in a table, use the following syntax:

```
ALTER TABLE table_name  
MODIFY COLUMN column_name data_type;
```



Drop the city column from vacation_plan table.

```
ALTER TABLE table_name  
DROP column_name;
```

▶ INSERT INTO



Syntax:

```
INSERT INTO table_name (column1, column2 ,...)  
VALUES( value1, value2 ,...);
```

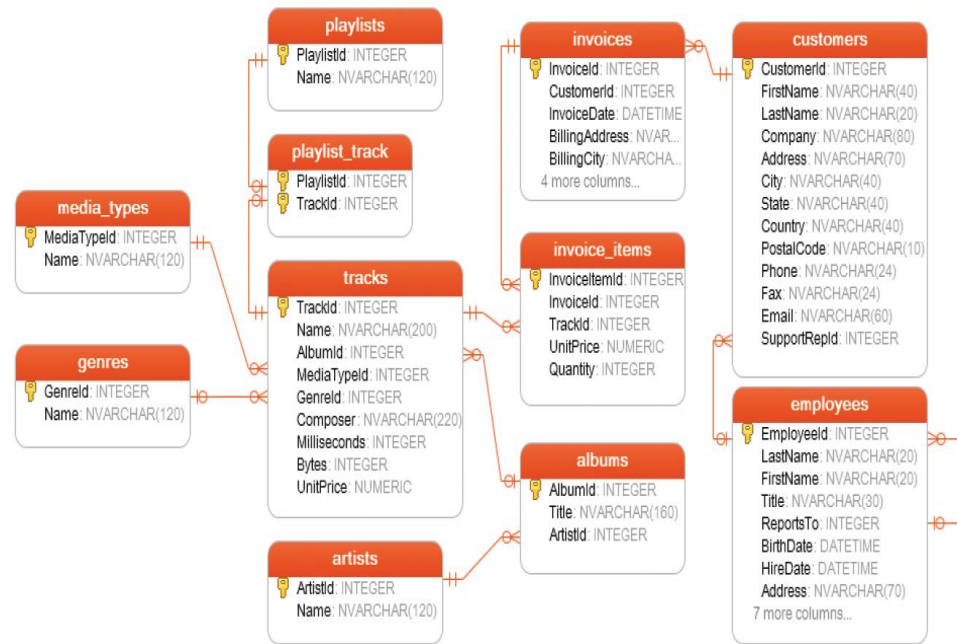
```
INSERT INTO table1 (column1,column2 ,...)  
VALUES  
(value1,value2 ,...),  
(value1,value2 ,...),  
...  
(value1,value2 ,...);
```



Query Time

INSERT a record for an employee into leaves table

id INT,
employee_id INT,
start_date DATE,
end_date DATE



Constraints



Constraints are the rules specified for data in a table. We can limit the type of data that will go into a table with the constraints. We can define the constraints with the **CREATE TABLE** statement or **ALTER TABLE** statement.

Constraints



Constraints

Constraint Name	Definition
NOT NULL	Ensures that a column cannot have a NULL value
DEFAULT	Sets a default value for a column when no value is specified
UNIQUE	Ensures that all values in a column are different
PRIMARY KEY	Uniquely identifies each row in a table
FOREIGN KEY	Uniquely identifies a row/record in another table

Primary Key



The primary key is a column in our table that makes each row (aka, record) unique.

Syntax

```
1 CREATE TABLE table_name(  
2     column_1 INT PRIMARY KEY,  
3     column_2 TEXT,  
4     ...  
5 );  
6 |
```


Primary Key



Syntax (Alternative)

```
1 CREATE TABLE table_name(  
2     column_1 INT,  
3     column_2 TEXT,  
4     ...  
5     PRIMARY KEY (column_1)  
6 );|
```

Foreign Key



Foreign key is a column in a table that uniquely identifies each row of another table. That column refers to a primary key of another table. This creates a kind of link between the tables.

Foreign Key



customers

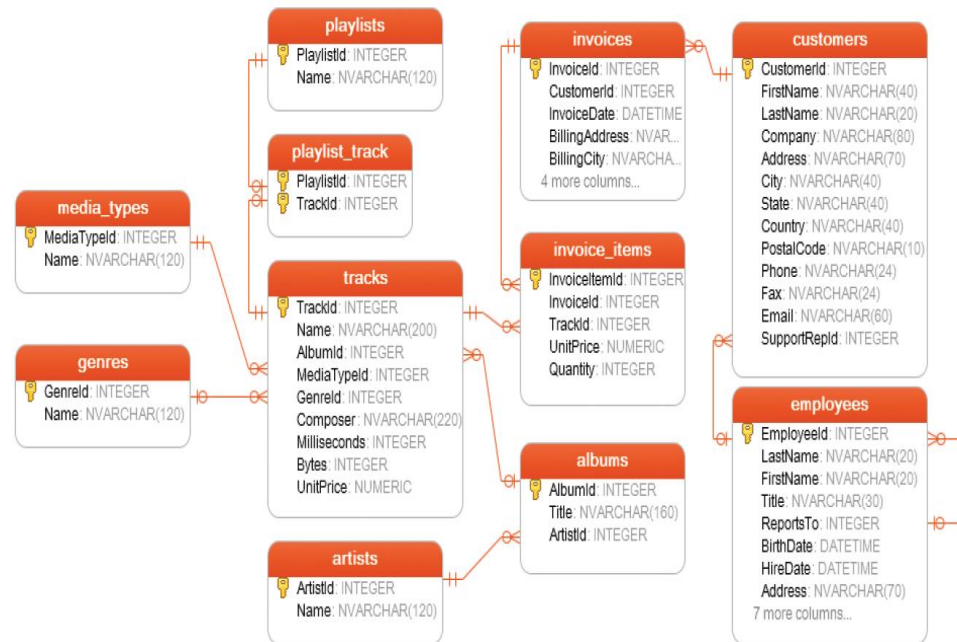
```
1 CREATE TABLE customers (customer_id INT PRIMARY KEY,  
2 first_name TEXT,  
3 second_name TEXT);  
4 |
```

orders

```
1 CREATE TABLE orders (  
2     order_id INT PRIMARY KEY,  
3     order_number INT,  
4     customer_id INT,  
5     FOREIGN KEY (customer_id)  
6     REFERENCES customers (customer_id)  
7 );  
8 |
```

Query Time

Try to insert a record in
albums table with an
ArtistID=10000 and
AlbumID=347



▶ Not Null



A column can include NULL values. A NULL value is a special value that means the value is unknown or does not exist.

All columns (except primary key's column) in a table can hold NULL values unless we explicitly specify **NOT NULL** constraints.

Not Null

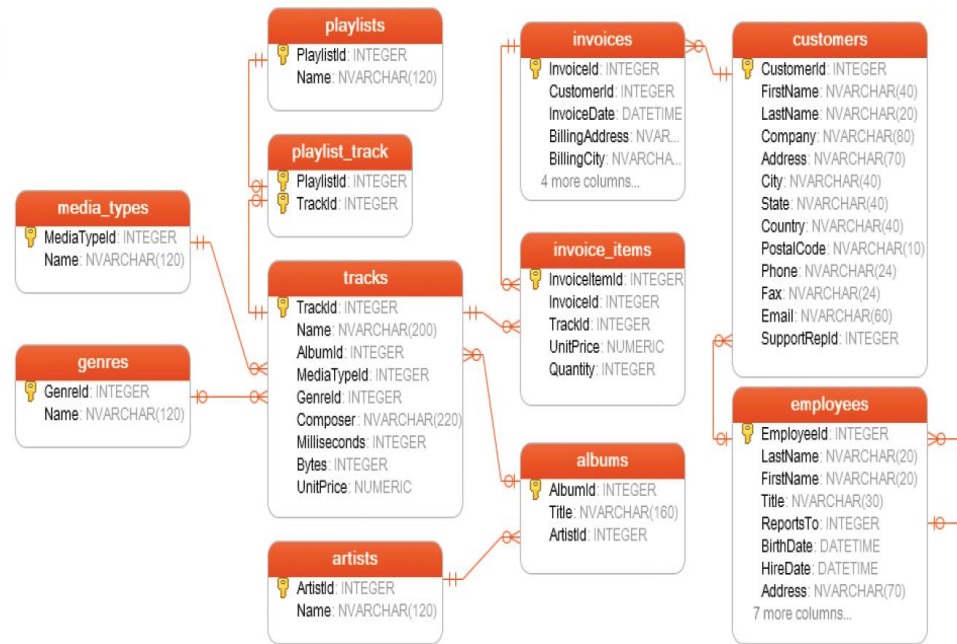


Syntax

```
1 CREATE TABLE table_name (  
2     column_name type_name NOT NULL,  
3     ...);  
4
```

Query Time

Try to insert a record in albums table without a title value





Please drop the table as you've just created writing

```
DROP TABLE vacation_plan;
```

Then, recreate the vacation_plan table adding constraints as below:

Column names:

- place_id -> PRIMARY KEY
- country
- hotel_name -> NOT NULL
- employee_id -> FOREIGN KEY
- vacation_length
- budget

Query Time



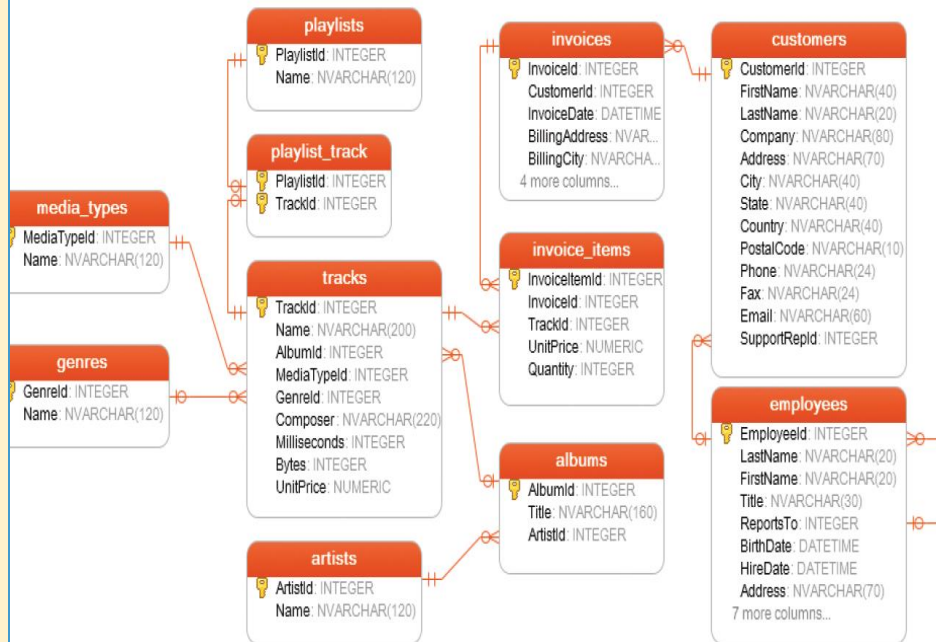
Please drop the table as you've just created writing

DROP TABLE leaves;

Then, recreate the **leaves** table adding constraints as below:

Column names:

- id -> PRIMARY KEY , AUTOINC
- employee_id -> FOREIGN KEY
- start_date -> NOT NULL
- end_date -> NOT NULL

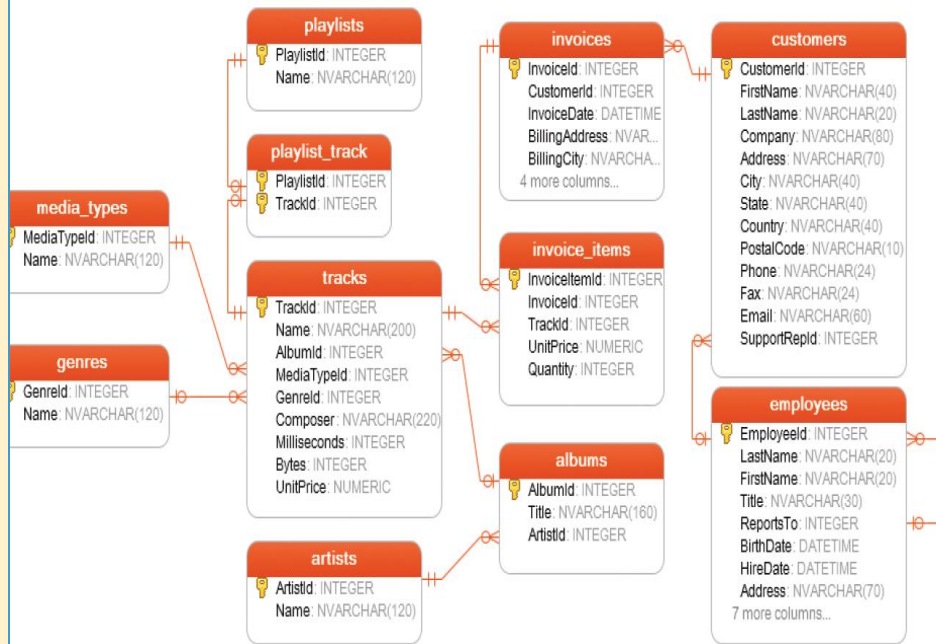


Query Time

Alter the table name to employee_leaves first. (Google this one know if you don't know)

Then add a column to your leaves table named "leave_type".

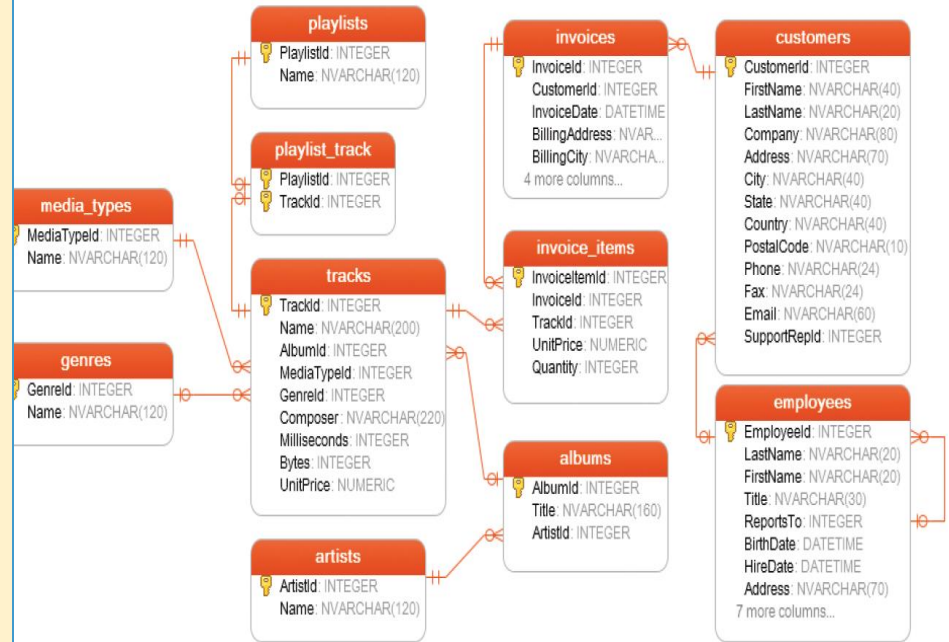
We will use type of leaves such as "annual leave", "sick leave" and etc.



Query Time

INSERT 3 new records to employee_leaves table.

You can use “annual_leave”, “sick_leave” and etc for leave type



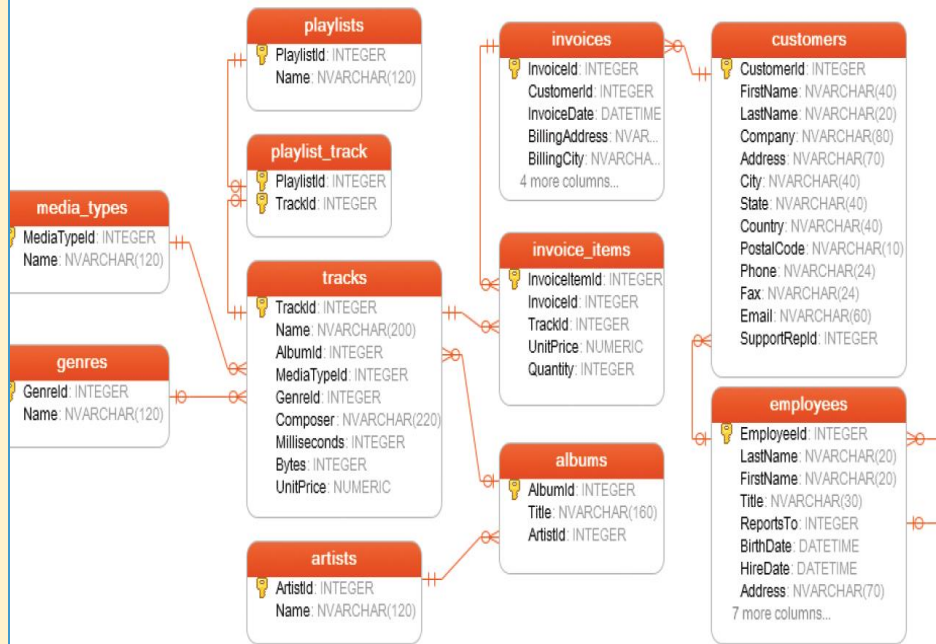
Query Time

Now add another table
leave_types with

id -> PK AUTOINC
leave_name -> TEXT

And make the column in
employee_leaves table as
FOREIGN KEY

ADD 3 records to leave_types
and employee_leaves table



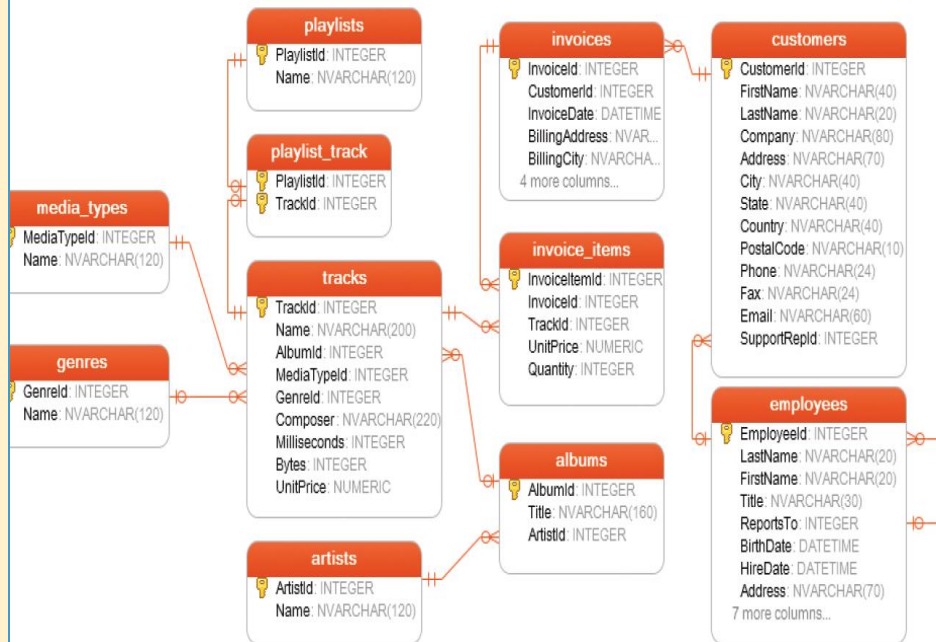
Query Time

Copy the entire table to a new table using csv import method

drop a column (you can't!)

change data type of a column

drop table



▶ UPDATE TABLE



```
UPDATE table  
  
SET column_1 = new_value_1,  
    column_2 = new_value_2  
  
WHERE  
    search_condition
```

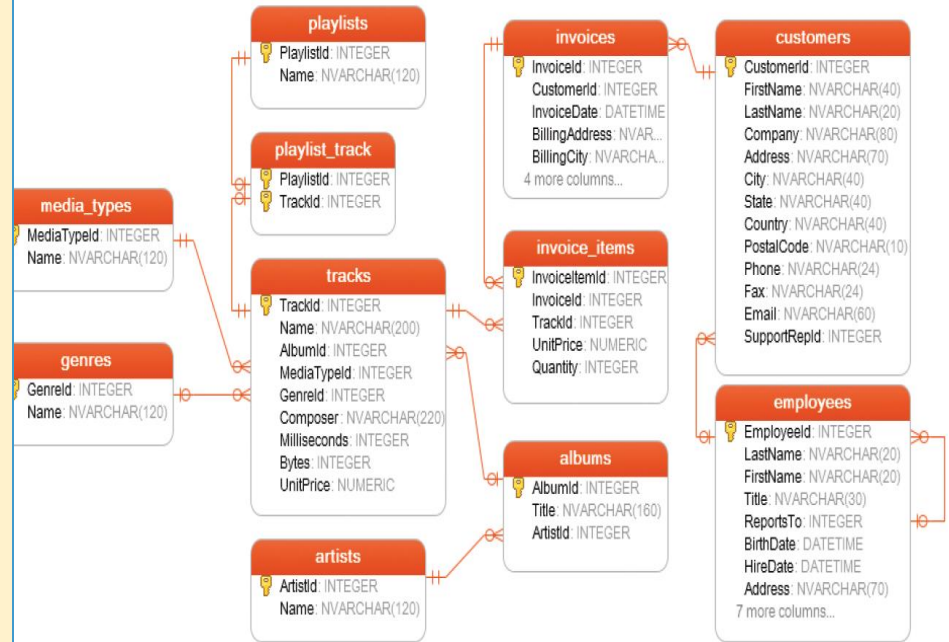
Query Time



Change the name of Annual leave to Marriage Leave in leave_types table

Change the start and end date values of a record in employee_leaves table

google sqlite date add



DELETE



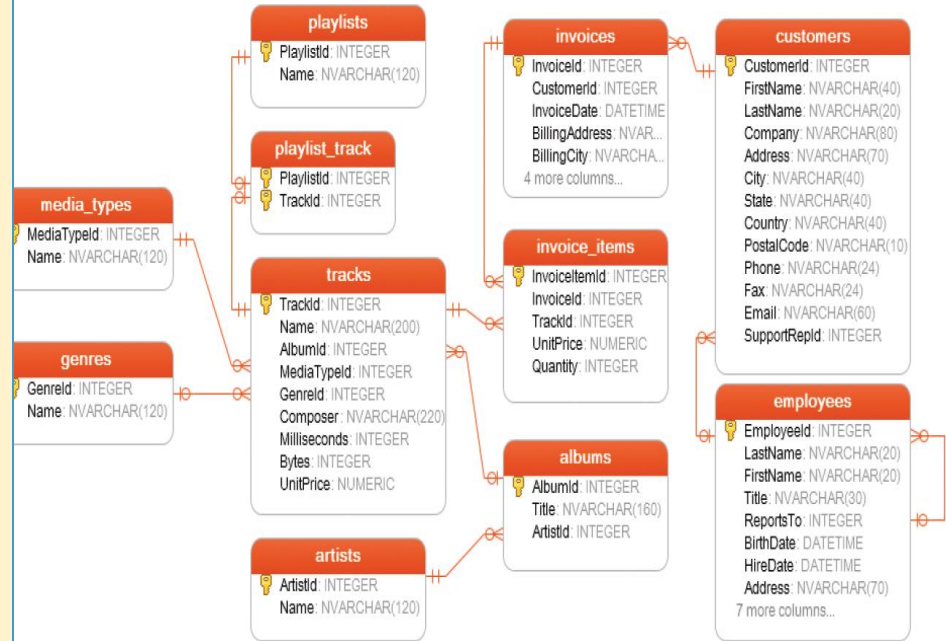
```
DELETE FROM table  
WHERE search_condition;
```


Query Time



Delete a record from leave types table

Delete a record from employee leaves table





Data Control Language



- Data Control Language (DCL) is used to grant or revoke access control.

GRANT

GRANT DELETE ON table_name TO user

REVOKE

REVOKE DELETE ON table_name FROM user



Data Control Language



```
GRANT SELECT,INSERT,UPDATE,DELETE ON *.* TO 'user_name'@'localhost';
```

```
GRANT ALL ON *.* TO 'user_name'@'localhost';
```

```
DROP USER user_name;
```



Index



- ▶ **CREATE INDEX**

```
CREATE INDEX indeks1 ON emp(id)
```

```
CREATE INDEX emp_idx1 ON emp(ename, job);
```

```
CREATE INDEX emp_idx2 ON emp(job, ename);
```

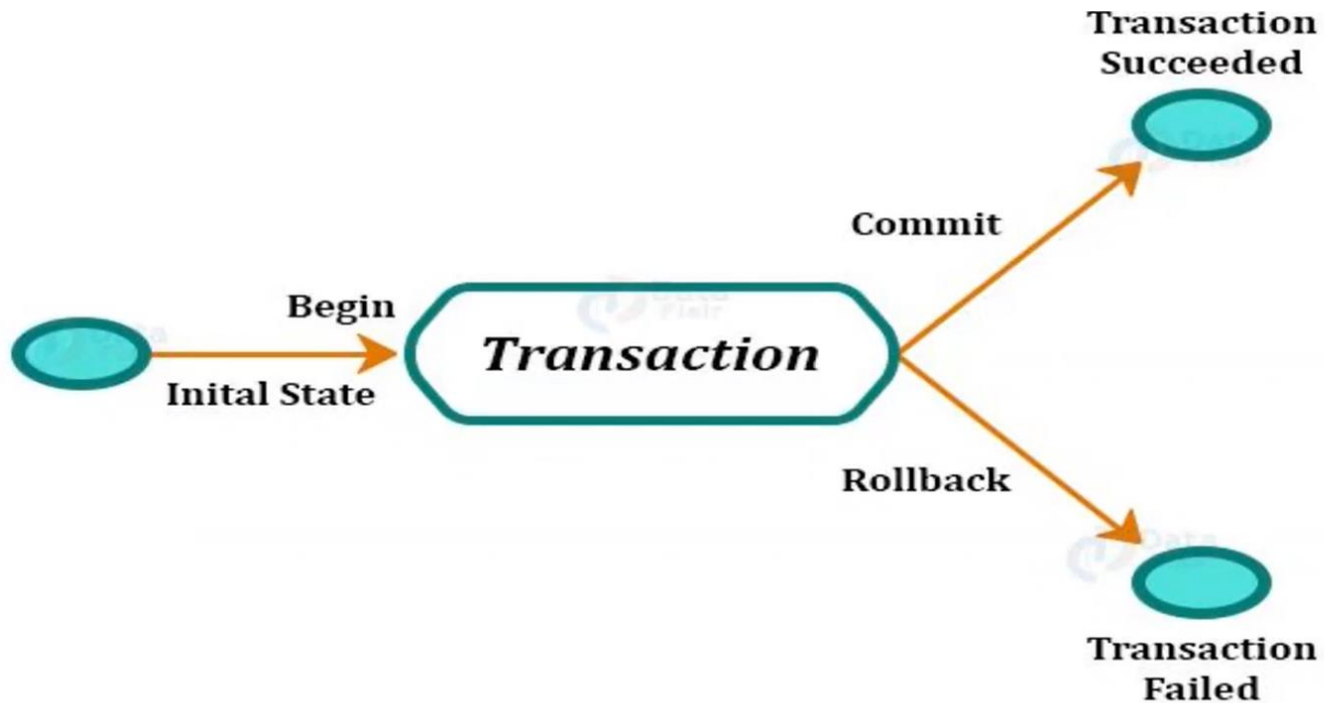


View



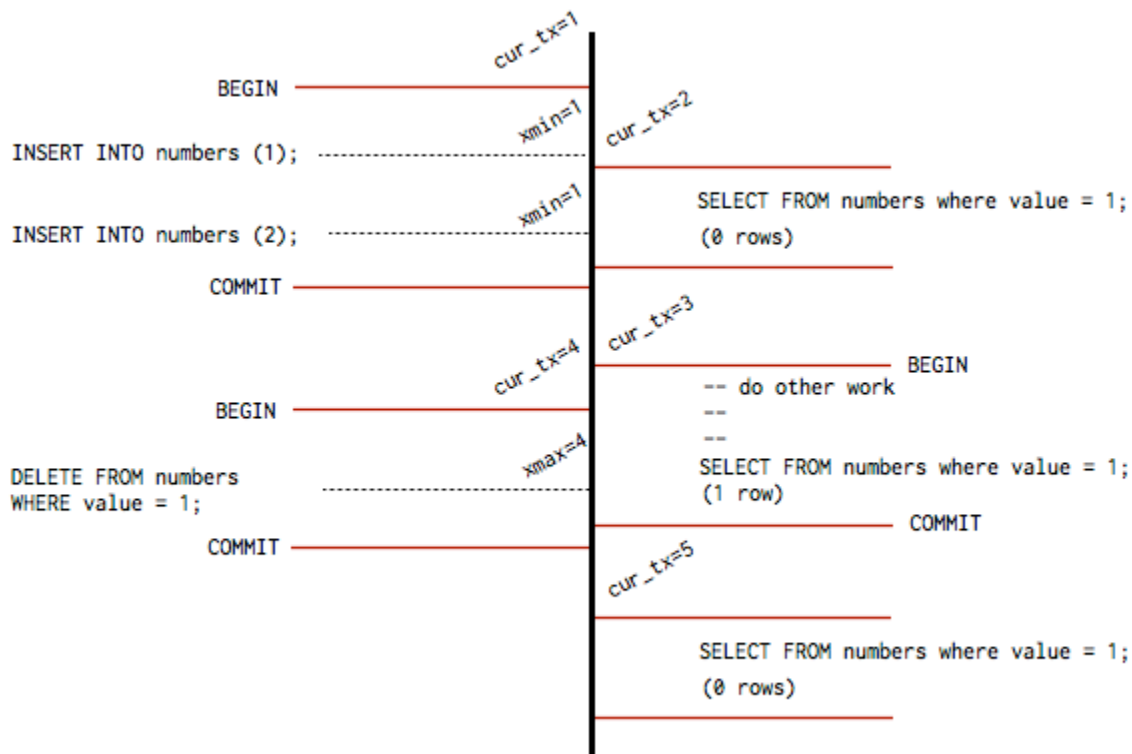
- ▶ `CREATE VIEW emp_view AS`
`SELECT empno, ename, sal, loc`
`FROM emp, dept`
`WHERE emp.deptno = dept.deptno`
`AND dept.deptno = 10;`
- ▶ `CREATE VIEW dept20 AS`
`SELECT ename, sal*12 annual_salary`
`FROM emp`
`WHERE deptno = 20;`

Transaction

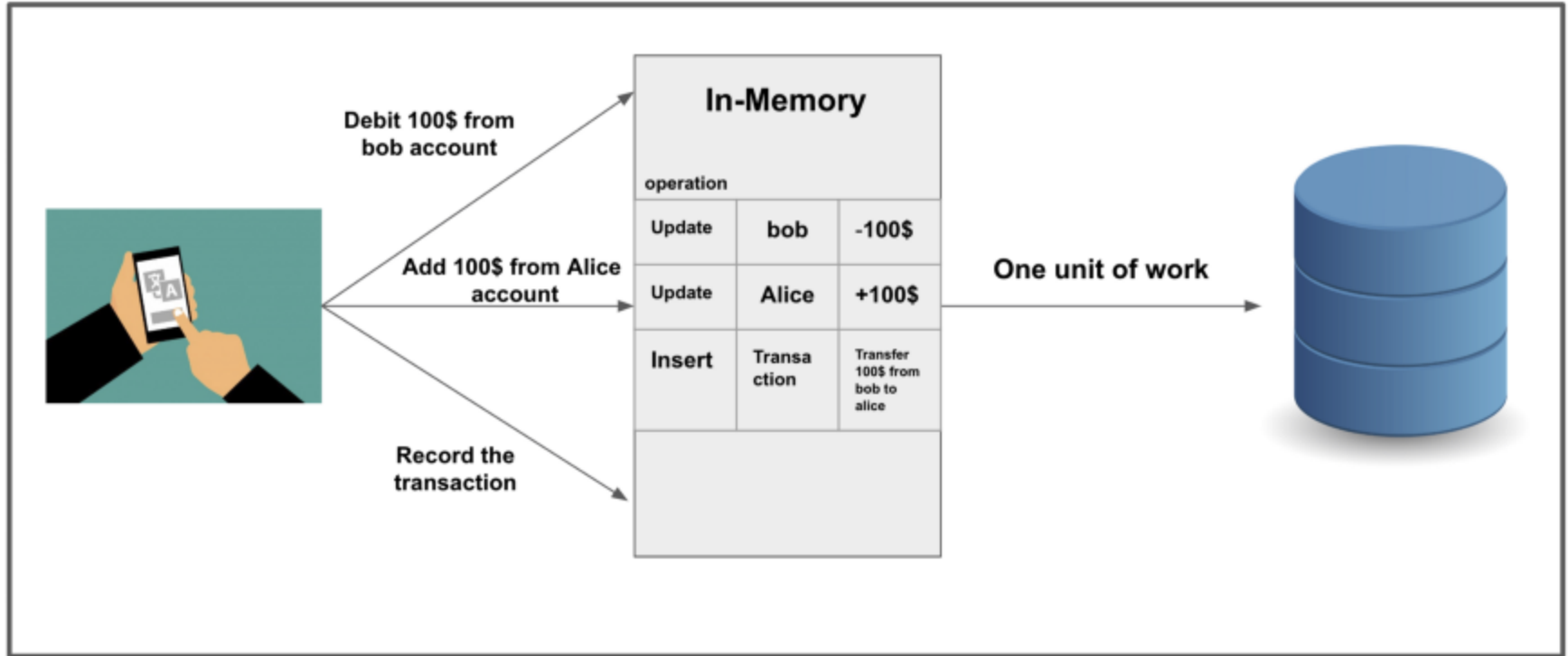




Transaction



Transaction





Transaction

```
mysql_connect("localhost", "username", "password");
```

```
mysql_select_db("db_name");
```

```
//transaction begining
```

```
mysql_query("BEGIN");
```

```
//query preparing
```

```
$query1 = mysql_query("UPDATE akbank SET acoount= acoount - 10000 WHERE account_no = '625021'");
```

```
$query2 = mysql_query("UPDATE garanti SET acoount = acoount + 10000 WHERE account_no = '124500'"); ,
```

```
if (!$query1 or !$query2 ){
```

```
    ("ROLLBACK");
```

```
}
```

```
else {
```

```
    ("COMMIT");
```

```
}
```



Trigger

```
CREATE [or REPLACE] TriggerName
[ BEFORE | AFTER ]
    [ DELETE | INSERT | UPDATE [of ColumnName ] ]
ON [User.]TableName
[ FOR EACH ROW ] [ WHEN Condition ]
BEGIN
    [PL/SQL Block]
END ;
```

THANKS!

