SQL

INFO6540
Week 8 - Mar 06, 2018

DALHOUSIE
UNIVERSITY
FACULTY OF MANAGEMENT
School of Information Management

Questions?

Assignment 1

- Read the questions

- Listen to what is said in class (details, explanations)

- Ask the instructor

- Look at the answers

- Re-open your saved file and make sure that all answers are there

- Make sure you submitted ALL the files

http://www.nationalgeographic.com.au/tv/meet-the-sloths/

Group Project

Content to be found in Brightspace under "Group Project".

Week 7

Relational databases & ER schemas

Data standards

Data modelling levels

- Conceptual: Very high level

- Logical: Models data requirements from an organization's point of view

- Physical: Model specific to a particular database technology

Relational schema

- Serves as a blueprint for data organization

DALHOUSIE
UNIVERSITY

FACULTY OF MANAGEMENT
School of Information Management
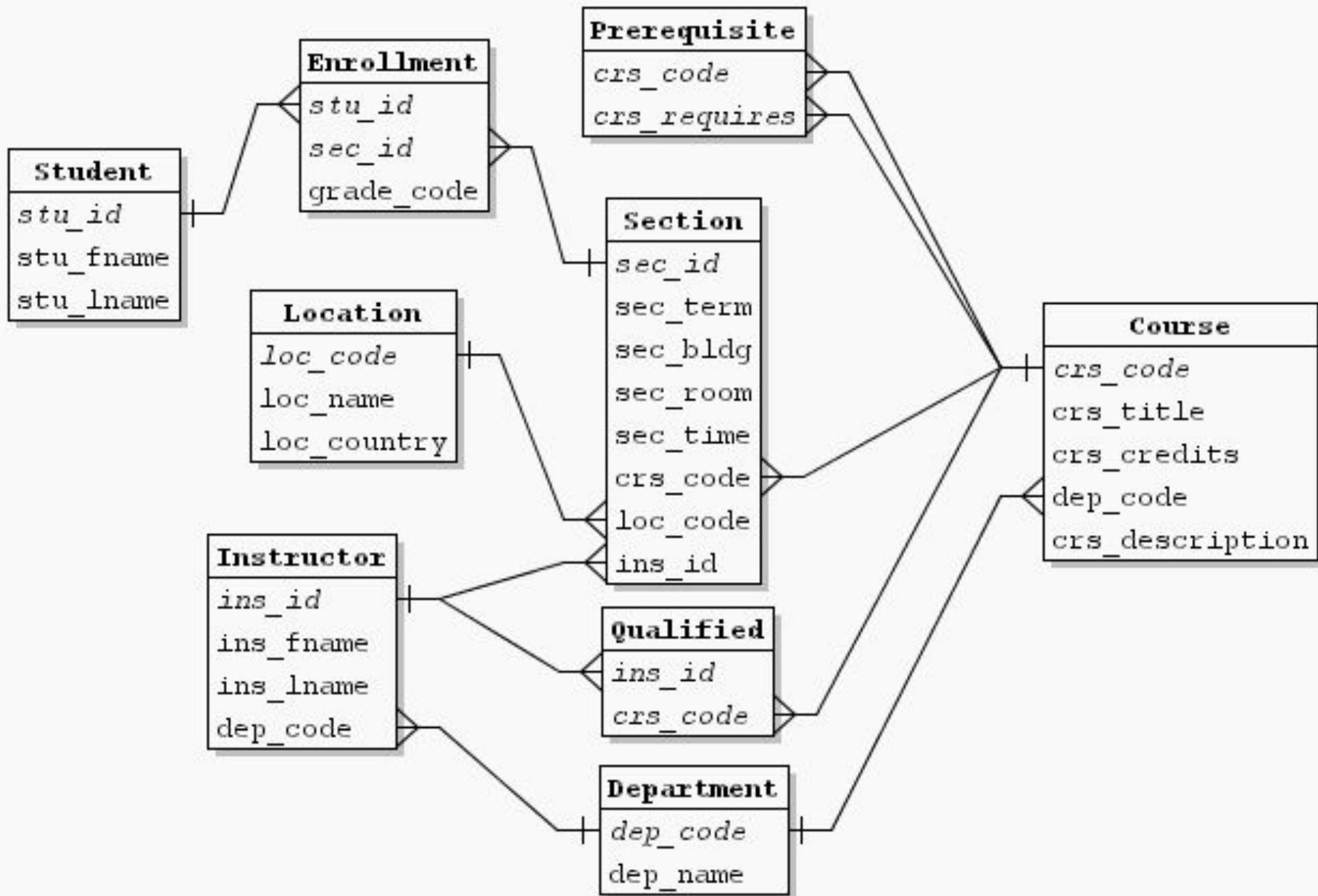
SQL

Structured Query Language

Standard language for querying and manipulating data.

A long time ago...

...we used to call it **SEQUEL**

D.D. Chamberlin and R.F. Boyce, "SEQUEL: A Structured English Query Language," in Proceedings of the 1974 ACM SIGFIDENT (now SIGMOD) workshop in Data Description, Access, and Control, New York, NY,ACM, 1974, pp. 249-264.

S~~EQU~~EL

- "set of simple operations on tabular structures"

- "bring the non-professional user into effective communication with a formatted database"

- "simplify programming and reduce the cost of software"

S~EQU~EL

- "There is also a large class of users who, while they are not computer specialists, would be willing to learn to interact with a computer in reasonably high-level, non-procedural query language."

SQL

- ## Data Definition Language (DDL)
  - ### Create/alter/delete tables

- ## Data Manipulation Language (DML)
  - ### Insert/update/delete data

	InvoiceLin	InvoiceID	TrackID	UnitPrice	Quantity
1	InvoiceLin	InvoiceID	TrackID	UnitPrice	Quantity
2	1	1	2	0.99	1
3	2	1	4	0.99	1
4	3	2	6	0.99	1
5	4	2	8	0.99	1
6	5	2	10	0.99	1
7	6	2	12	0.99	1
8	7	3	16	0.99	1
9	8	3	20	0.99	1
10	9	3	24	0.99	1
11	10	3	28	0.99	1
12	11	3	32	0.99	1
13	12	3	36	0.99	1
14	13	4	42	0.99	1
15	14	4	48	0.99	1
16	15	4	54	0.99	1
17	16	4	60	0.99	1
18	17	4	66	0.99	1
19	18	4	72	0.99	1
20	19	4	78	0.99	1

invoice_items

	InvoiceID	Customer|	Date	BillingAddress	BillingiCity	BillingStat|	BillingCou|	PostalCod|	Total
1	InvoiceID	Customer|	Date	BillingAddress	BillingiCity	BillingStat|	BillingCou|	PostalCod|	Total
2	1	2	1/1/09 12:00 AM	Theodor-Heuss-Straße 34	Stuttgart		Germany	70174	1.98
3	2	4	1/2/2009 0:00	Ullevålsveien 14	Oslo		Norway	171	3.96
4	3	8	1/3/2009 0:00	Grétrystraat 63	Brussels		Belgium	1000	5.94
5	4	14	1/6/2009 0:00	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	8.91
6	5	23	1/11/2009 0:00	69 Salem Street	Boston	MA	USA	2113	13.86
7	6	37	1/19/2009 0:00	Berger Straße 10	Frankfurt		Germany	60316	0.99
8	7	38	2/1/2009 0:00	Barbarossastraße 19	Berlin		Germany	10779	1.98
9	8	40	2/1/2009 0:00	8, Rue Hanovre	Paris		France	75002	1.98
10	9	42	2/2/2009 0:00	9, Place Louis Barthou	Bordeaux		France	33000	3.96
11	10	46	2/3/2009 0:00	3 Chatham Street	Dublin	Dublin	Ireland		5.94
12	11	52	2/6/2009 0:00	202 Hoxton Street	London		United Kin	N1 5LH	8.91
13	12	2	2/11/2009 0:00	Theodor-Heuss-Straße 34	Stuttgart		Germany	70174	13.86
14	13	16	2/19/2009 0:00	1600 Amphitheatre Parkw	Mountain View	CA	USA	94043-135	0.99
15	14	17	3/4/2009 0:00	1 Microsoft Way	Redmond	WA	USA	98052-830	1.98
16	15	19	3/4/2009 0:00	1 Infinite Loop	Cupertino	CA	USA	95014	1.98
17	16	21	3/5/2009 0:00	801 W 4th Street	Reno	NV	USA	89503	3.96
18	17	25	3/6/2009 0:00	319 N. Frances Street	Madison	WI	USA	53703	5.94
19	18	31	3/9/2009 0:00	194A Chain Lake Drive	Halifax	NS	Canada	B3S 1C5	8.91
20	19	40	3/14/2009 0:00	8, Rue Hanovre	Paris		France	75002	13.86

invoices

	TrackID	Name	AlbumID	MediaTyp	GenreID	Composer	Millisecon	Bytes	UnitPrice
1	TrackID	Name	AlbumID	MediaTyp	GenreID	Composer	Millisecon	Bytes	UnitPrice
2	1	For Those About To Rock (We Sal	1	1	1	Angus Young, Malcolm Young	343719	11170334	0.99
3	2	Balls to the Wall	2	2	1		342562	5510424	0.99
4	3	Fast As a Shark	3	2	1	F. Baltes, S. Kaufman, U. Dirks	230619	3990994	0.99
5	4	Restless and Wild	3	2	1	F. Baltes, R.A. Smith-Diesel, S.	252051	4331779	0.99
6	5	Princess of the Dawn	3	2	1	Deaffy & R.A. Smith-Diesel	375418	6290521	0.99
7	6	Put The Finger On You	1	1	1	Angus Young, Malcolm Young	205662	6713451	0.99
8	7	Let's Get It Up	1	1	1	Angus Young, Malcolm Young	233926	7636561	0.99
9	8	Inject The Venom	1	1	1	Angus Young, Malcolm Young	210834	6852860	0.99
10	9	Snowballed	1	1	1	Angus Young, Malcolm Young	203102	6599424	0.99
11	10	Evil Walks	1	1	1	Angus Young, Malcolm Young	263497	8611245	0.99
12	11	C.O.D.	1	1	1	Angus Young, Malcolm Young	199836	6566314	0.99
13	12	Breaking The Rules	1	1	1	Angus Young, Malcolm Young	263288	8596840	0.99
14	13	Night Of The Long Knives	1	1	1	Angus Young, Malcolm Young	205688	6706347	0.99
15	14	Spellbound	1	1	1	Angus Young, Malcolm Young	270863	8817038	0.99
16	15	Go Down	4	1	1	AC/DC	331180	10847611	0.99
17	16	Dog Eat Dog	4	1	1	AC/DC	215196	7032162	0.99
18	17	Let There Be Rock	4	1	1	AC/DC	366654	12021261	0.99
19	18	Bad Boy Boogie	4	1	1	AC/DC	267728	8776140	0.99
20	19	Problem Child	4	1	1	AC/DC	325041	10617116	0.99

tracks

SQLITE TUTORIAL

playlists
- 🔑 PlaylistId: INTEGER
- Name: NVARCHAR(120)

playlist_track
- 🔑 PlaylistId: INTEGER
- 🔑 TrackId: INTEGER

media_types
- 🔑 MediaTypeId: INTEGER
- Name: NVARCHAR(120)

genres
- 🔑 GenreId: INTEGER
- Name: NVARCHAR(120)

tracks
- 🔑 TrackId: INTEGER
- Name: NVARCHAR(200)
- AlbumId: INTEGER
- MediaTypeId: INTEGER
- GenreId: INTEGER
- Composer: NVARCHAR(220)
- Milliseconds: INTEGER
- Bytes: INTEGER
- UnitPrice: NUMERIC

artists
- 🔑 ArtistId: INTEGER
- Name: NVARCHAR(120)

invoices
- 🔑 InvoiceId: INTEGER
- CustomerId: INTEGER
- InvoiceDate: DATETIME
- BillingAddress: NVAR...
- BillingCity: NVARCHA...
- 4 more columns...

invoice_items
- 🔑 InvoiceItemId: INTEGER
- InvoiceId: INTEGER
- TrackId: INTEGER
- UnitPrice: NUMERIC
- Quantity: INTEGER

albums
- 🔑 AlbumId: INTEGER
- Title: NVARCHAR(160)
- ArtistId: INTEGER

customers
- 🔑 CustomerId: INTEGER
- FirstName: NVARCHAR(40)
- LastName: NVARCHAR(20)
- Company: NVARCHAR(80)
- Address: NVARCHAR(70)
- City: NVARCHAR(40)
- State: NVARCHAR(40)
- Country: NVARCHAR(40)
- PostalCode: NVARCHAR(10)
- Phone: NVARCHAR(24)
- Fax: NVARCHAR(24)
- Email: NVARCHAR(60)
- SupportRepId: INTEGER

employees
- 🔑 EmployeeId: INTEGER
- LastName: NVARCHAR(20)
- FirstName: NVARCHAR(20)
- Title: NVARCHAR(30)
- ReportsTo: INTEGER
- BirthDate: DATETIME
- HireDate: DATETIME
- Address: NVARCHAR(70)
- 7 more columns...

- 'employees' table stores employee data such as employee id, last name, first name, etc. It also has a field named 'ReportsTo' to specify who reports to whom.
- 'customers' table stores customer data
- 'invoices' & 'invoice_items' tables: these two tables store invoice data. The 'invoices' table stores invoice header data and the 'invoice_items' table stores the invoice line items data.
- 'artists' table stores artists data. It is a simple table that contains only artist id and name.
- 'albums_table' stores data about a list of tracks. Each album belongs to one artist. However, one artist may have multiple albums.

- 'media_types' table stores media types such as MPEG audio file, ACC audio file, etc.
- 'genres_table' genres table stores music types such as rock, jazz, metal, etc.
- 'tracks table' stores the data of songs. Each track belongs to one album.
- 'playlists' & 'playlist track' tables: 'playlists' table store data about playlists. Each playlist contains a list of tracks. Each track may belong to multiple playlists. The relationship between the 'playlists' table and 'tracks' table is many-to-many. The 'playlist_track' table is used to reflect this relationship.

Data types

- CHAR(size): Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis.
- nVARCHAR(size): Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis.
- TEXT: Holds a string with a maximum length of 65,535 characters.

Data types

- INTEGER: Whole numbers.
- NUMERIC: Decimal numbers.
- DATE: A date. Format: YYYY-MM-DD.
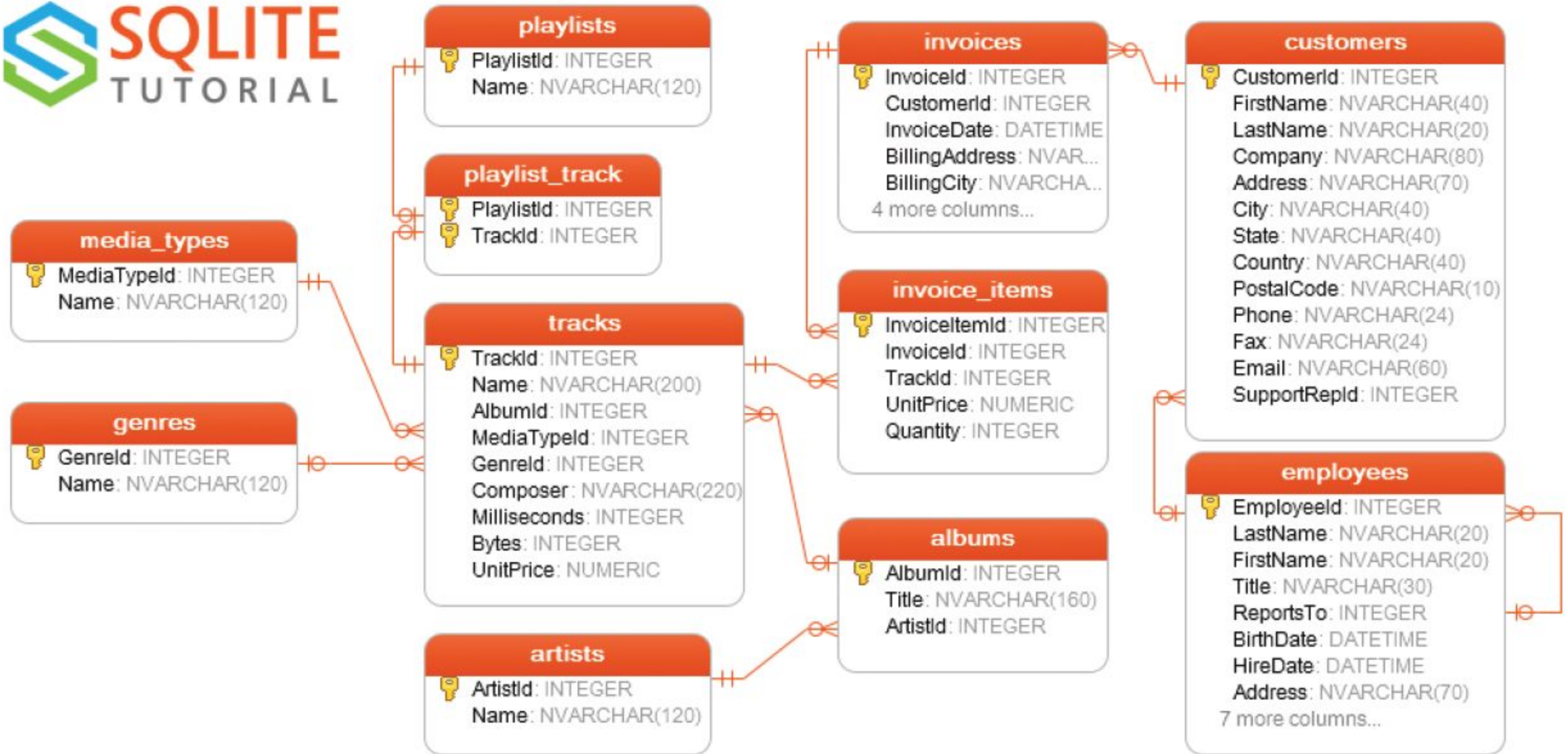- DATETIME: A date and time combination. Format: YYYY-MM-DD HH:MI:SS.
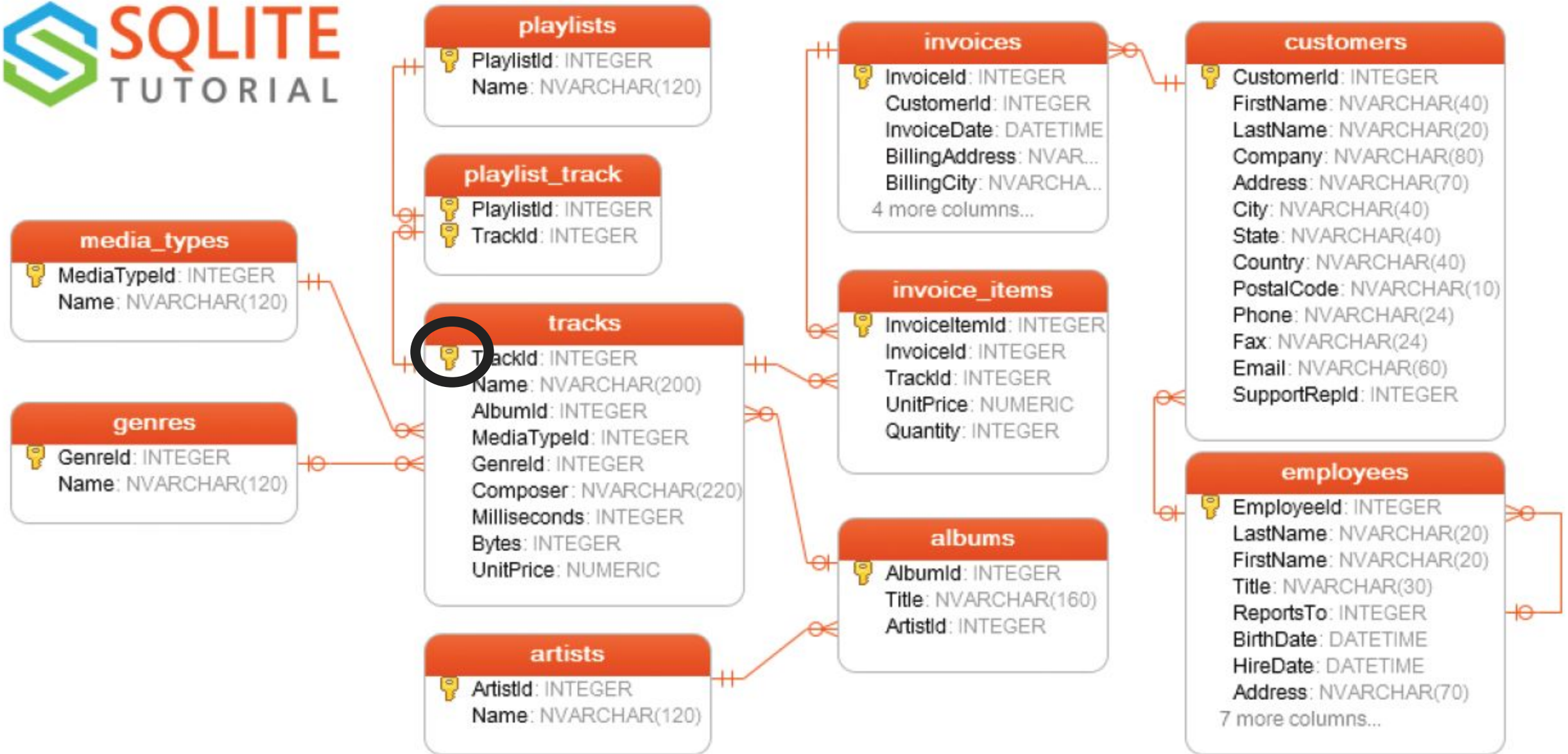
Primary key

- Consists of one or more columns whose data within is used to uniquely identify each row in the table.

- No values in the columns can be blank or NULL.

- Just one per table.

Foreign key

- Consists of one or more columns in a table that refers to the primary key in another table.

- No need for "special code", configuration, or table definition to designate it.

SQLITE TUTORIAL

playlists
- 🔑 PlaylistId: INTEGER
- Name: NVARCHAR(120)

playlist_track
- 🔑 PlaylistId: INTEGER
- 🔑 TrackId: INTEGER

media_types
- 🔑 MediaTypeId: INTEGER
- Name: NVARCHAR(120)

genres
- 🔑 GenreId: INTEGER
- Name: NVARCHAR(120)

tracks
- 🔑 TrackId: INTEGER
- Name: NVARCHAR(200)
- AlbumId: INTEGER
- MediaTypeId: INTEGER
- GenreId: INTEGER
- Composer: NVARCHAR(220)
- Milliseconds: INTEGER
- Bytes: INTEGER
- UnitPrice: NUMERIC

artists
- 🔑 ArtistId: INTEGER
- Name: NVARCHAR(120)

invoices
- 🔑 InvoiceId: INTEGER
- CustomerId: INTEGER
- InvoiceDate: DATETIME
- BillingAddress: NVAR...
- BillingCity: NVARCHA...
- 4 more columns...

invoice_items
- 🔑 InvoiceItemId: INTEGER
- InvoiceId: INTEGER
- TrackId: INTEGER
- UnitPrice: NUMERIC
- Quantity: INTEGER

albums
- 🔑 AlbumId: INTEGER
- Title: NVARCHAR(160)
- ArtistId: INTEGER

customers
- 🔑 CustomerId: INTEGER
- FirstName: NVARCHAR(40)
- LastName: NVARCHAR(20)
- Company: NVARCHAR(80)
- Address: NVARCHAR(70)
- City: NVARCHAR(40)
- State: NVARCHAR(40)
- Country: NVARCHAR(40)
- PostalCode: NVARCHAR(10)
- Phone: NVARCHAR(24)
- Fax: NVARCHAR(24)
- Email: NVARCHAR(60)
- SupportRepId: INTEGER

employees
- 🔑 EmployeeId: INTEGER
- LastName: NVARCHAR(20)
- FirstName: NVARCHAR(20)
- Title: NVARCHAR(30)
- ReportsTo: INTEGER
- BirthDate: DATETIME
- HireDate: DATETIME
- Address: NVARCHAR(70)
- 7 more columns...

FACULTY OF MANAGEMENT
School of Information Management

Basic commands

- Data Definition Language (DDL)
  - Create/alter/delete tables

- Data Manipulation Language (DML)
  - Insert/update/delete/query data

DDL

- CREATE - to create objects in the database
- ALTER - alters the structure of the database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
- RENAME - rename an object

DML

- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain
- JOIN - combines rows from two or more tables based on a related column between them

CREATE

CREATE TABLE table_name (
    column_1 datatype column_constraint,
    column_2 datatype column_constraint,
   column_3 datatype column_constraint,
  ....
);

CREATE

CREATE TABLE contacts (
 contact_id integer PRIMARY KEY,
 first_name text NOT NULL,
 last_name text NOT NULL,
 email text NOT NULL UNIQUE,
 phone text NOT NULL UNIQUE
);

CREATE

CREATE TABLE contact_groups (
 contact_id integer,
 group_id integer,
 PRIMARY KEY (contact_id, group_id),
 FOREIGN KEY (contact_id) REFERENCES contacts (contact_id),
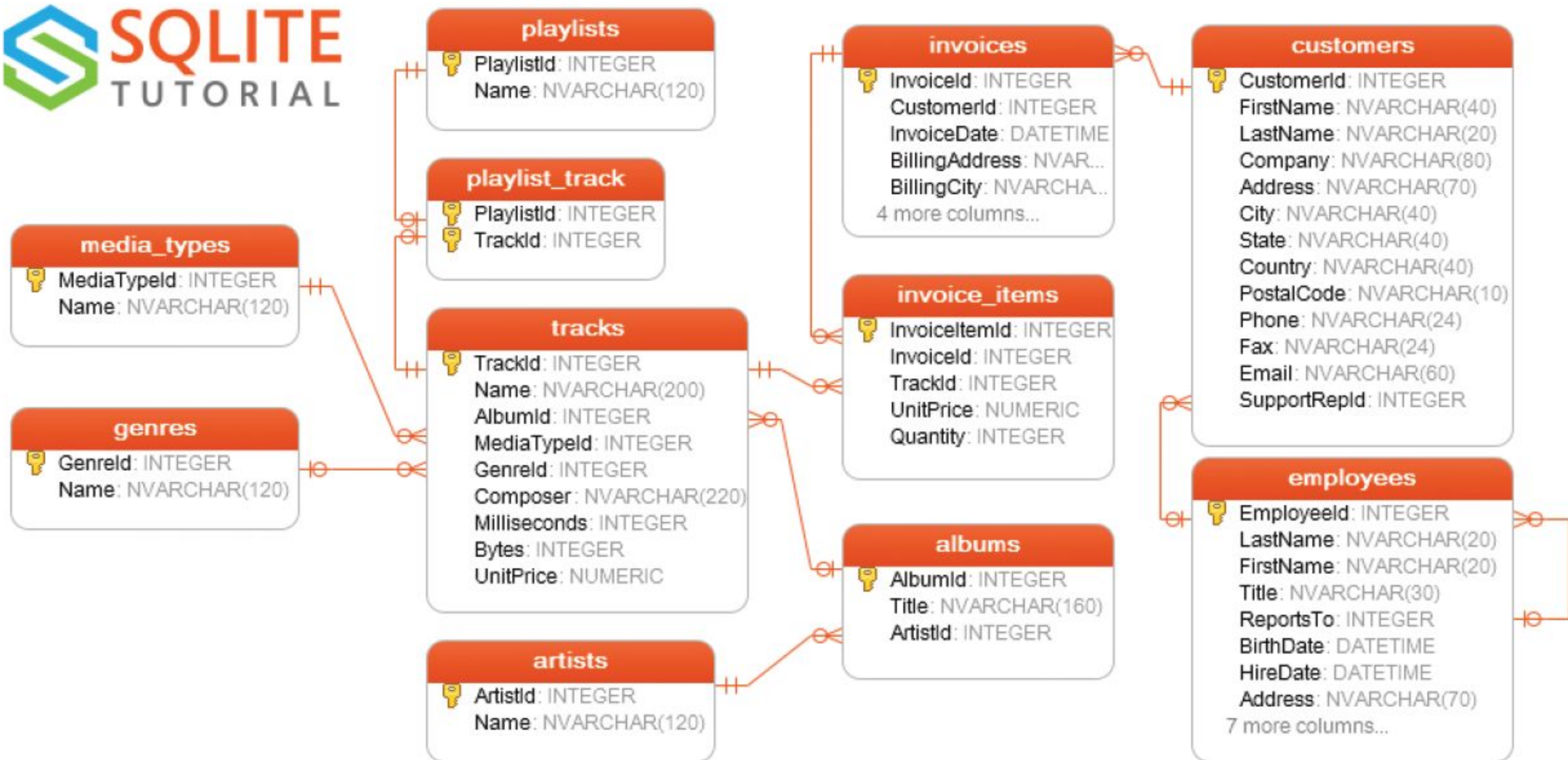 FOREIGN KEY (group_id) REFERENCES groups (group_id)
 );

groups
- 🔑 group_id: integer
- name: text

contact_groups
- 🔑 contact_id: integer
- 🔑 group_id: integer

contacts
- 🔑 contact_id: integer
- first_name: text
- last_name: text
- email: text
- phone: text
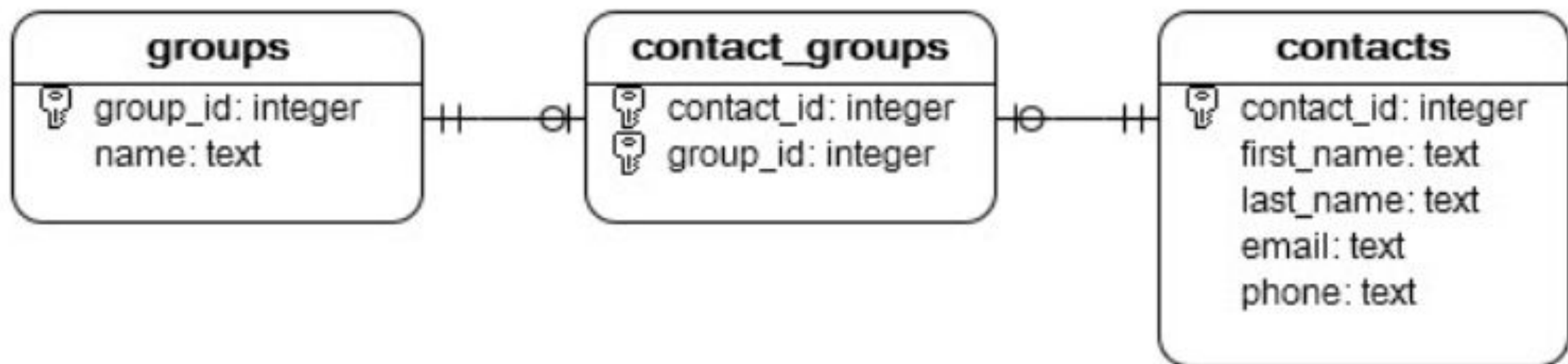
INSERT

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

INSERT

INSERT INTO groups (
group_id,
name)
VALUES
(
"001",
"Europeans");

SQLiteStudio

DALHOUSIE
UNIVERSITY
FACULTY OF MANAGEMENT
School of Information Management