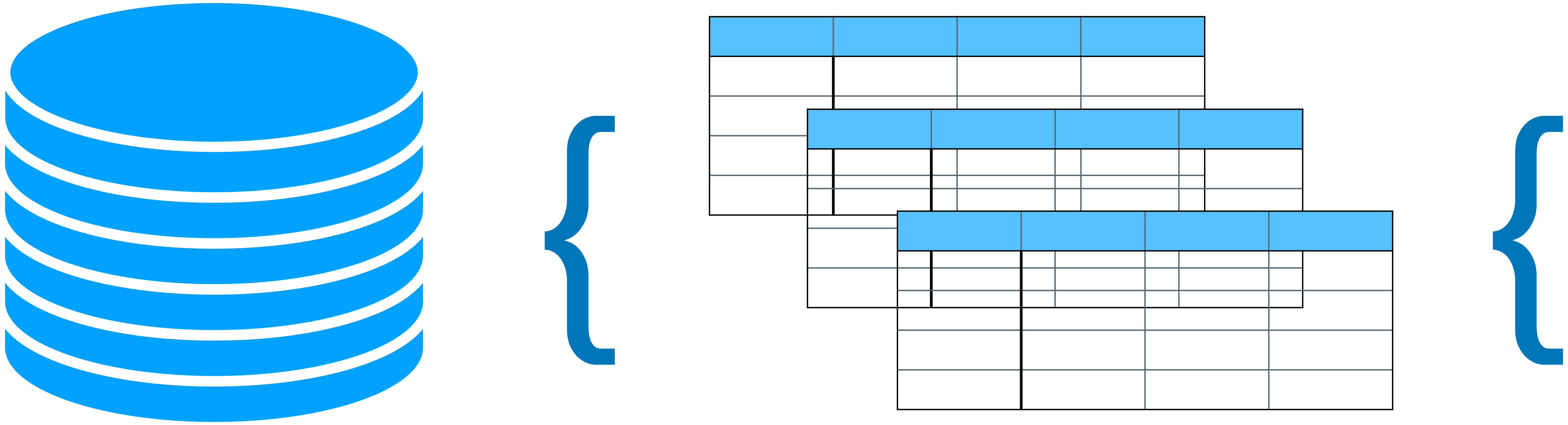
Data definition language (DDL) & constraints

Database design

NIFR 2024



Database / Schema

Tables

Rows and columns (data)

Database: MySQL

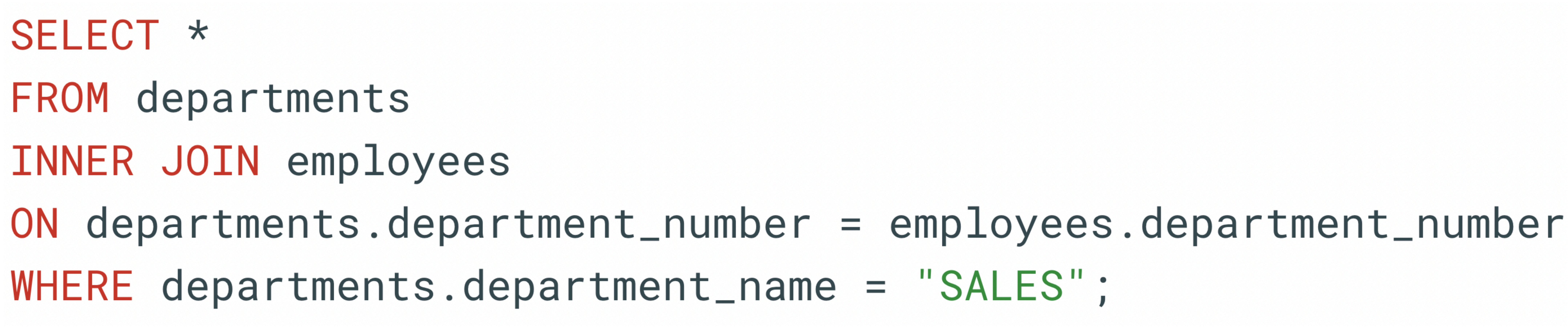
Language: SQL

DATA DEFINITION LANGUAGE

DATA QUERY LANGUAGE

DATA MODELLING LANGUAGE

Data Query [Manipulation] Language



```sql
SELECT *
FROM departments
INNER JOIN employees
ON departments.department_number = employees.department_number
WHERE departments.department_name = "SALES";
```

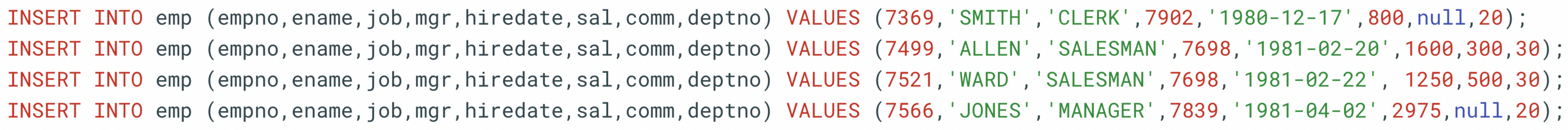
Data Query Language

Query > Response (Data)

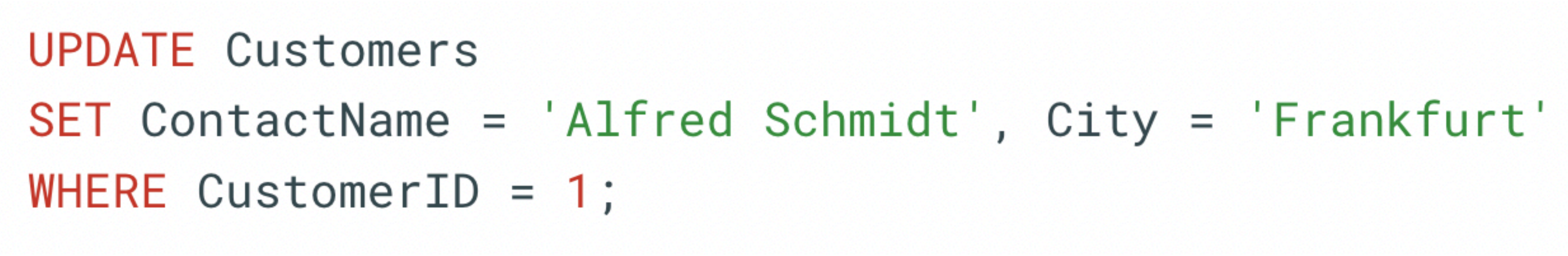
Structured Query Language (SQL)

- Data Definition Language (DDL)
- Data Modelling Language (DML)
- Data Query Language (DQL)

Data Modelling Language



```
INSERT INTO emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) VALUES (7369,'SMITH','CLERK',7902,'1980-12-17',800,null,20);
INSERT INTO emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) VALUES (7499,'ALLEN','SALESMAN',7698,'1981-02-20',1600,300,30);
INSERT INTO emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) VALUES (7521,'WARD','SALESMAN',7698,'1981-02-22', 1250,500,30);
INSERT INTO emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) VALUES (7566,'JONES','MANAGER',7839,'1981-04-02',2975,null,20);
```



```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City = 'Frankfurt'
WHERE CustomerID = 1;
```

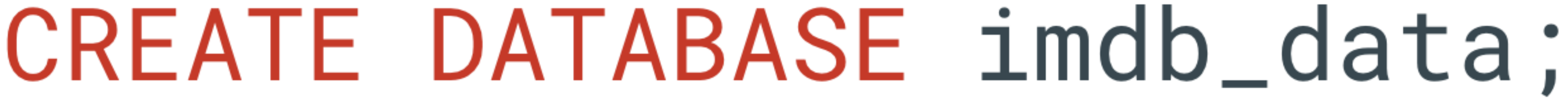
Data Modelling Language

- INSERT INTO
  - Insert data into table
- DELETE
  - Delete data from table
- UPDATE
  - Update data from table

Data Modelling Language

Query > Success or Failure

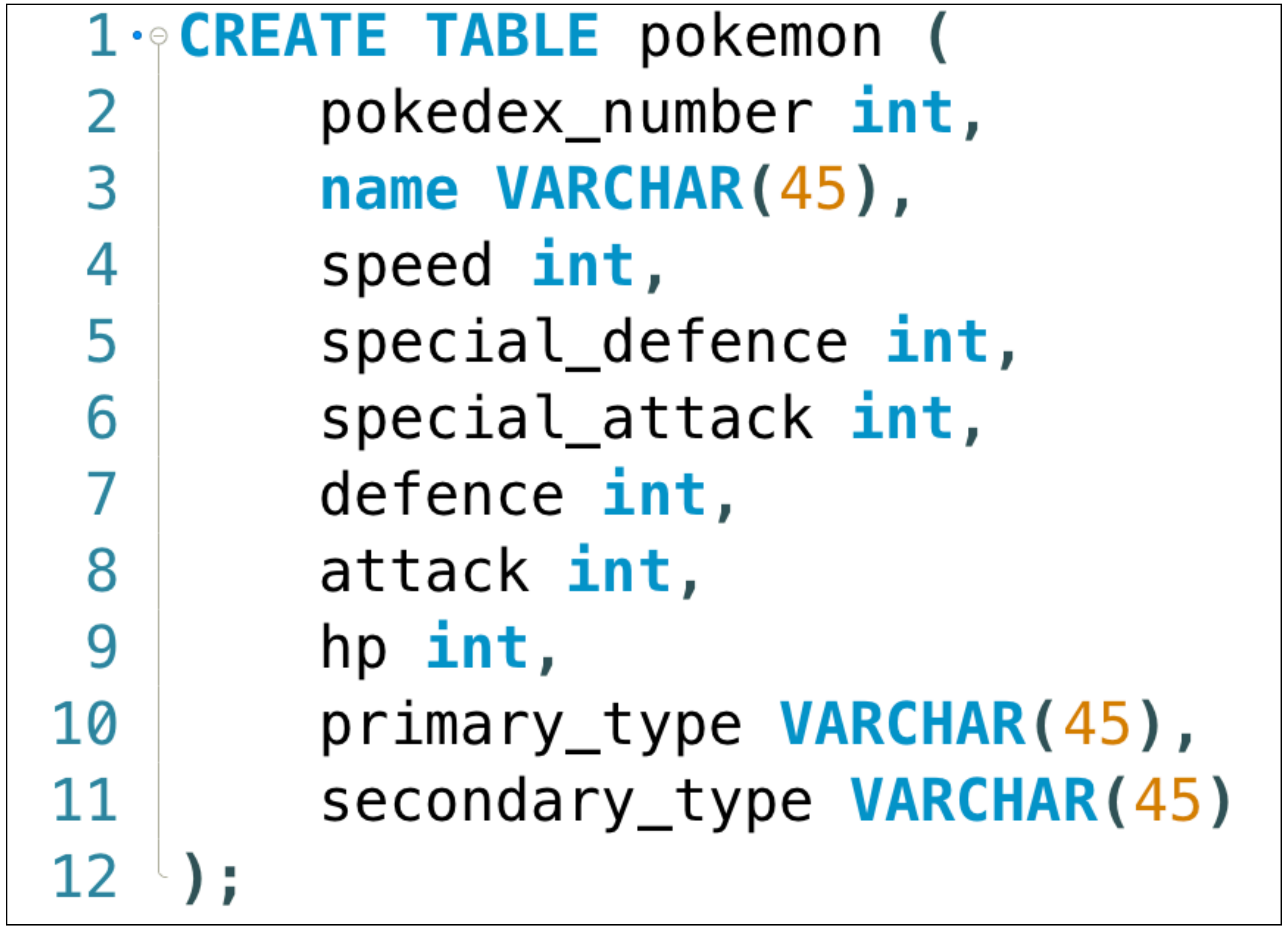
Data Definition Language



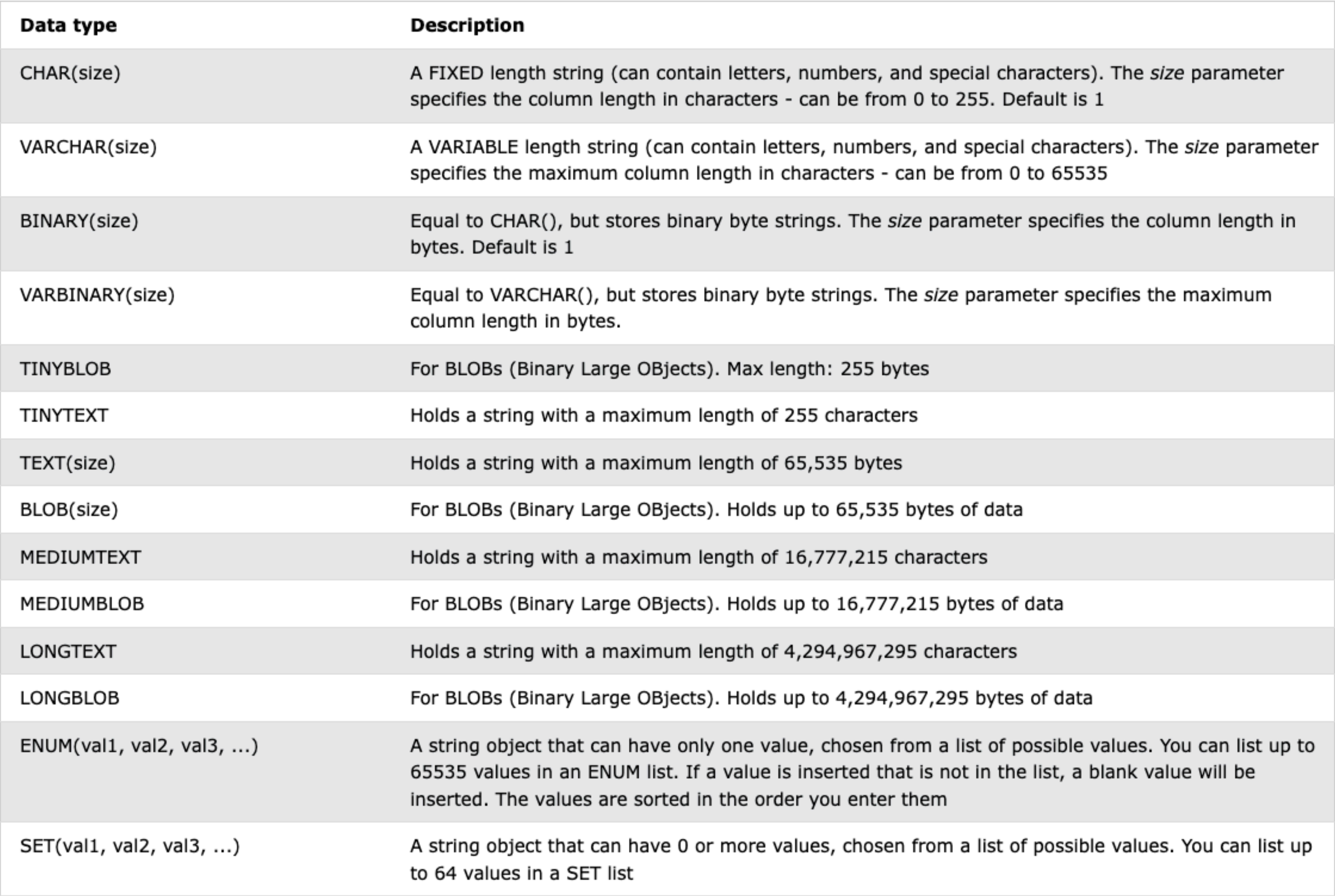
```
CREATE DATABASE imdb_data;
```

Data Definition Language DDL

- CREATE: Create database/table
- ALTER: Change table e.g. add a column
- DROP: Delete table
- TRUNCATE: Delete all data in the table, keep the table



```sql
CREATE TABLE pokemon (
	pokedex_number int,
	name VARCHAR(45),
	speed int,
	special_defence int,
	special_attack int,
	defence int,
	attack int,
	hp int,
	primary_type VARCHAR(45),
	secondary_type VARCHAR(45)
);
```

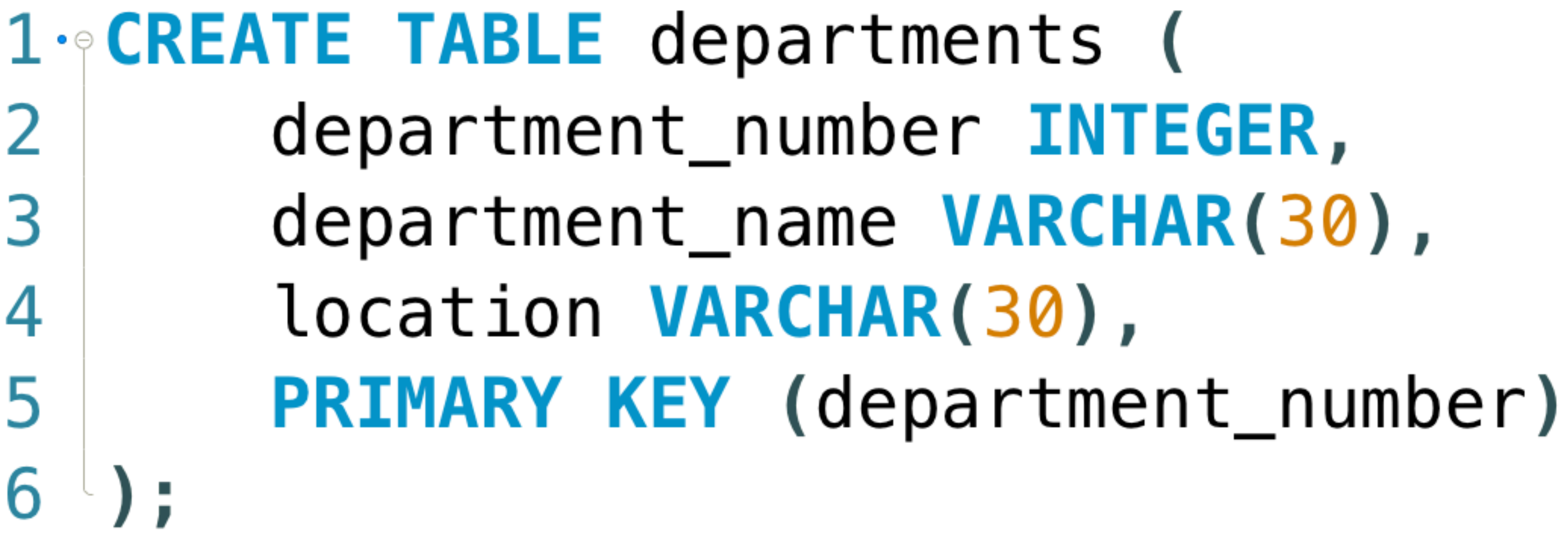


Data type	Description
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters). The *size* parameter specifies the column length in characters - can be from 0 to 255. Default is 1
VARCHAR(size)	A VARIABLE length string (can contain letters, numbers, and special characters). The *size* parameter specifies the maximum column length in characters - can be from 0 to 65535
BINARY(size)	Equal to CHAR(), but stores binary byte strings. The *size* parameter specifies the column length in bytes. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. The *size* parameter specifies the maximum column length in bytes.
TINYBLOB	For BLOBs (Binary Large OBjects). Max length: 255 bytes
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT(size)	Holds a string with a maximum length of 65,535 bytes
BLOB(size)	For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LONGBLOB	For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data
ENUM(val1, val2, val3, ...)	A string object that can have only one value, chosen from a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. The values are sorted in the order you enter them
SET(val1, val2, val3, ...)	A string object that can have 0 or more values, chosen from a list of possible values. You can list up to 64 values in a SET list

https://www.w3schools.com/mysql/mysql_datatypes.asp

Constraints DDL

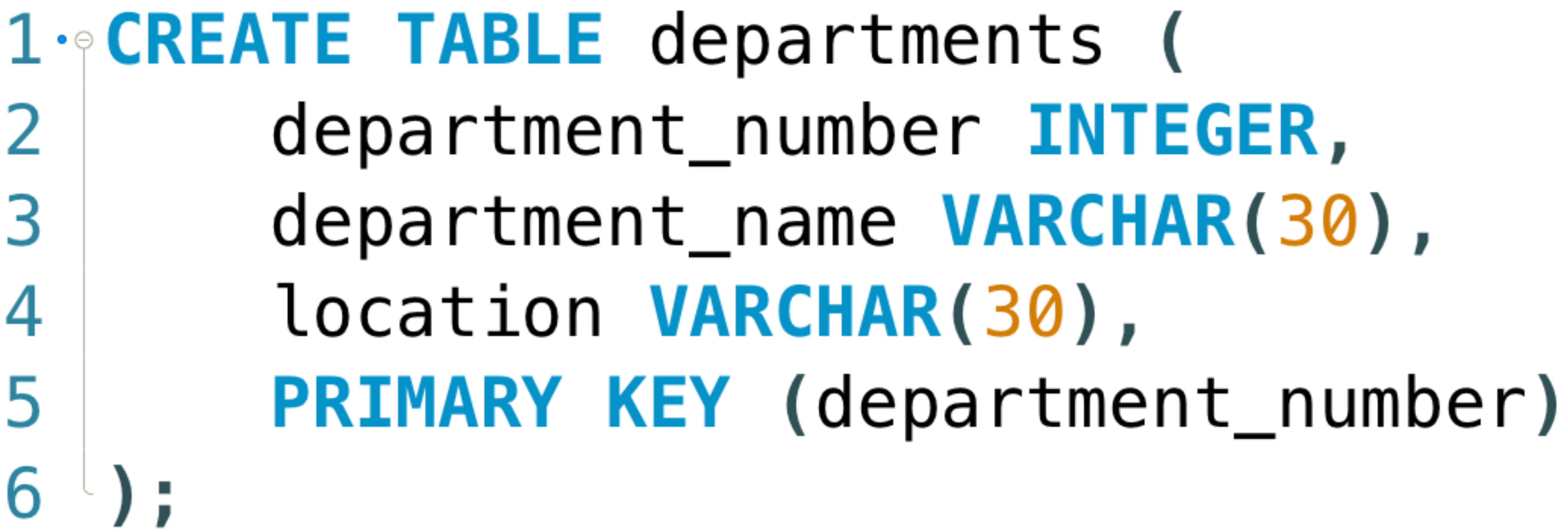
- PRIMARY KEY
- FOREIGN KEY
- DEFAULT
- NOT NULL



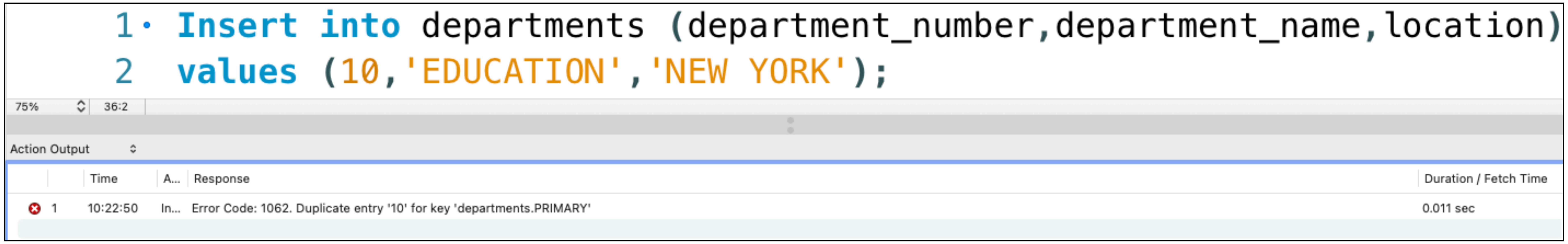
```sql
CREATE TABLE departments (
      department_number INTEGER,
      department_name VARCHAR(30),
      location VARCHAR(30),
      PRIMARY KEY (department_number)
);
```

Constraints DDL

- Department_number is constrained
- Now the data inserted has to be unique & cannot be null



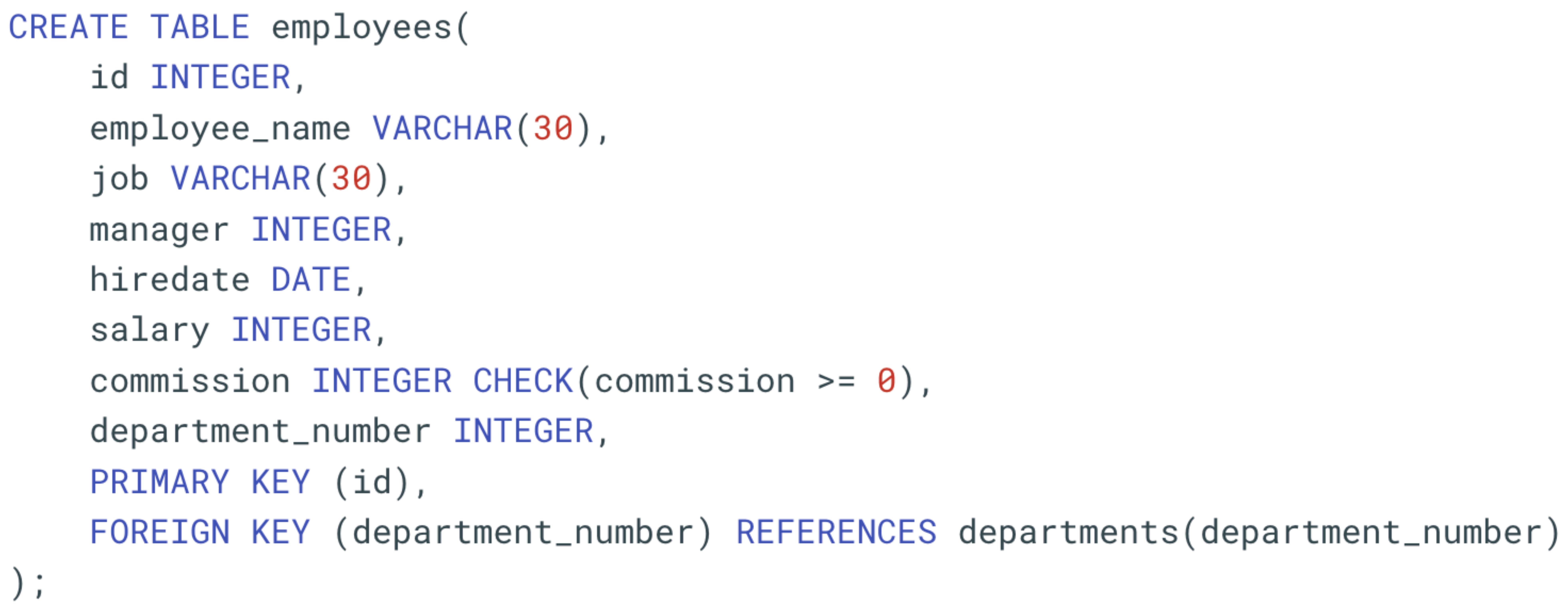
```
CREATE TABLE departments (
      department_number INTEGER,
      department_name VARCHAR(30),
      location VARCHAR(30),
      PRIMARY KEY (department_number)
);
```



```
Insert into departments (department_number,department_name,location)
values (10,'EDUCATION','NEW YORK');
```

```
    Time      A...  Response                                                            Duration / Fetch Time
 1  10:22:50  In... Error Code: 1062. Duplicate entry '10' for key 'departments.PRIMARY'   0.011 sec
```

Check Constraint



```
CREATE TABLE employees(
    id INTEGER,
    employee_name VARCHAR(30),
    job VARCHAR(30),
    manager INTEGER,
    hiredate DATE,
    salary INTEGER,
    commission INTEGER CHECK(commission >= 0),
    department_number INTEGER,
    PRIMARY KEY (id),
    FOREIGN KEY (department_number) REFERENCES departments(department_number)
);
```

https://www.mysqltutorial.org/mysql-check-constraint/

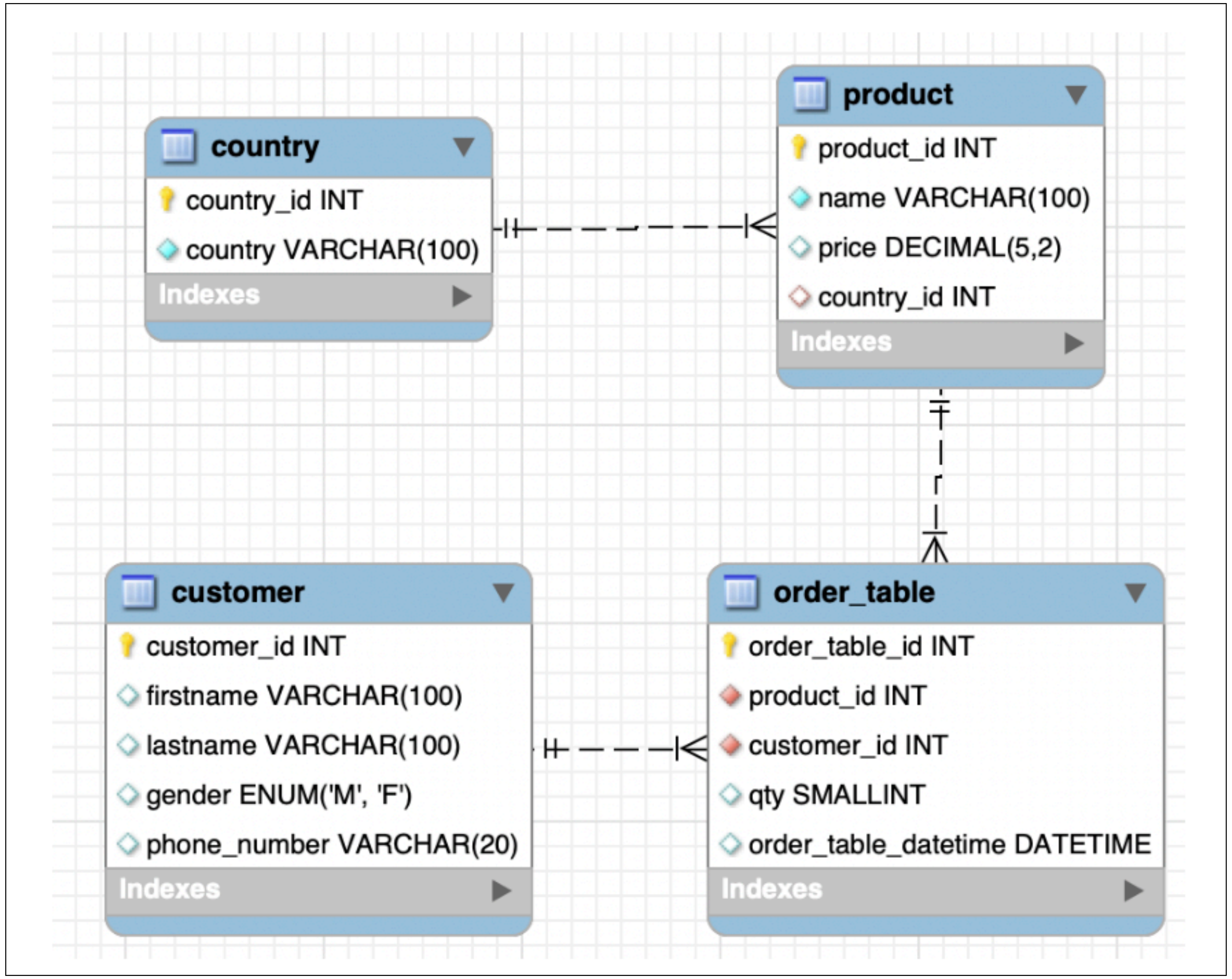
Check Constraint

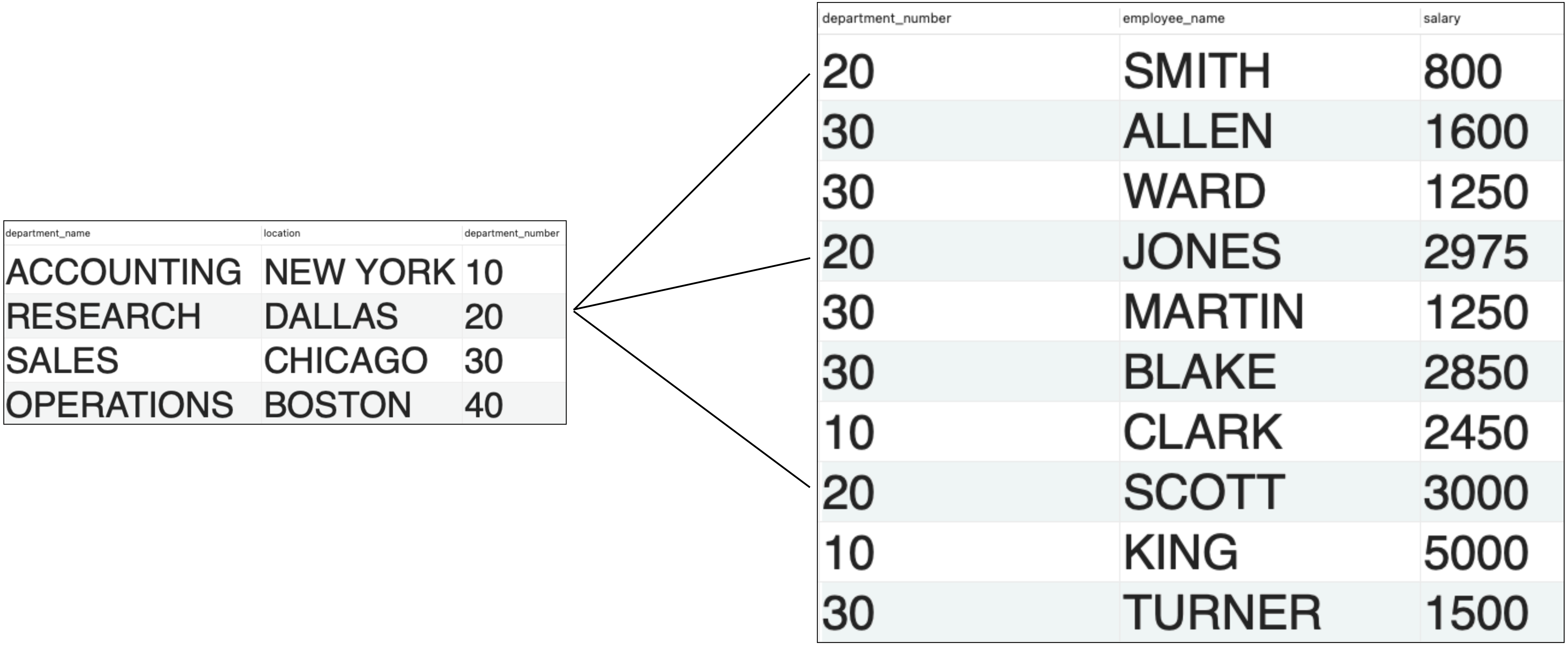
```
CREATE TABLE employees(
    id INTEGER,
    employee_name VARCHAR(30),
    job VARCHAR(30),
    manager INTEGER,
    hiredate DATE,
    salary INTEGER,
    commission INTEGER CHECK(commission >= 0),
    department_number INTEGER,
    PRIMARY KEY (id),
    FOREIGN KEY (department_number) REFERENCES departments(department_number)
);
```

https://www.mysqltutorial.org/mysql-check-constraint/

Exercise 1

What defines the relational model?


country
country_id INT
country VARCHAR(100)
Indexes
product
product_id INT
name VARCHAR(100)
price DECIMAL(5,2)
country_id INT
Indexes
customer
customer_id INT
firstname VARCHAR(100)
lastname VARCHAR(100)
gender ENUM('M', 'F')
phone_number VARCHAR(20)
Indexes
order_table
order_table_id INT
product_id INT
customer_id INT
qty SMALLINT
order_table_datetime DATETIME
Indexes



department_name	location	department_number
ACCOUNTING	NEW YORK	10
RESEARCH	DALLAS	20
SALES	CHICAGO	30
OPERATIONS	BOSTON	40

department_number	employee_name	salary
20	SMITH	800
30	ALLEN	1600
30	WARD	1250
20	JONES	2975
30	MARTIN	1250
30	BLAKE	2850
10	CLARK	2450
20	SCOTT	3000
10	KING	5000
30	TURNER	1500

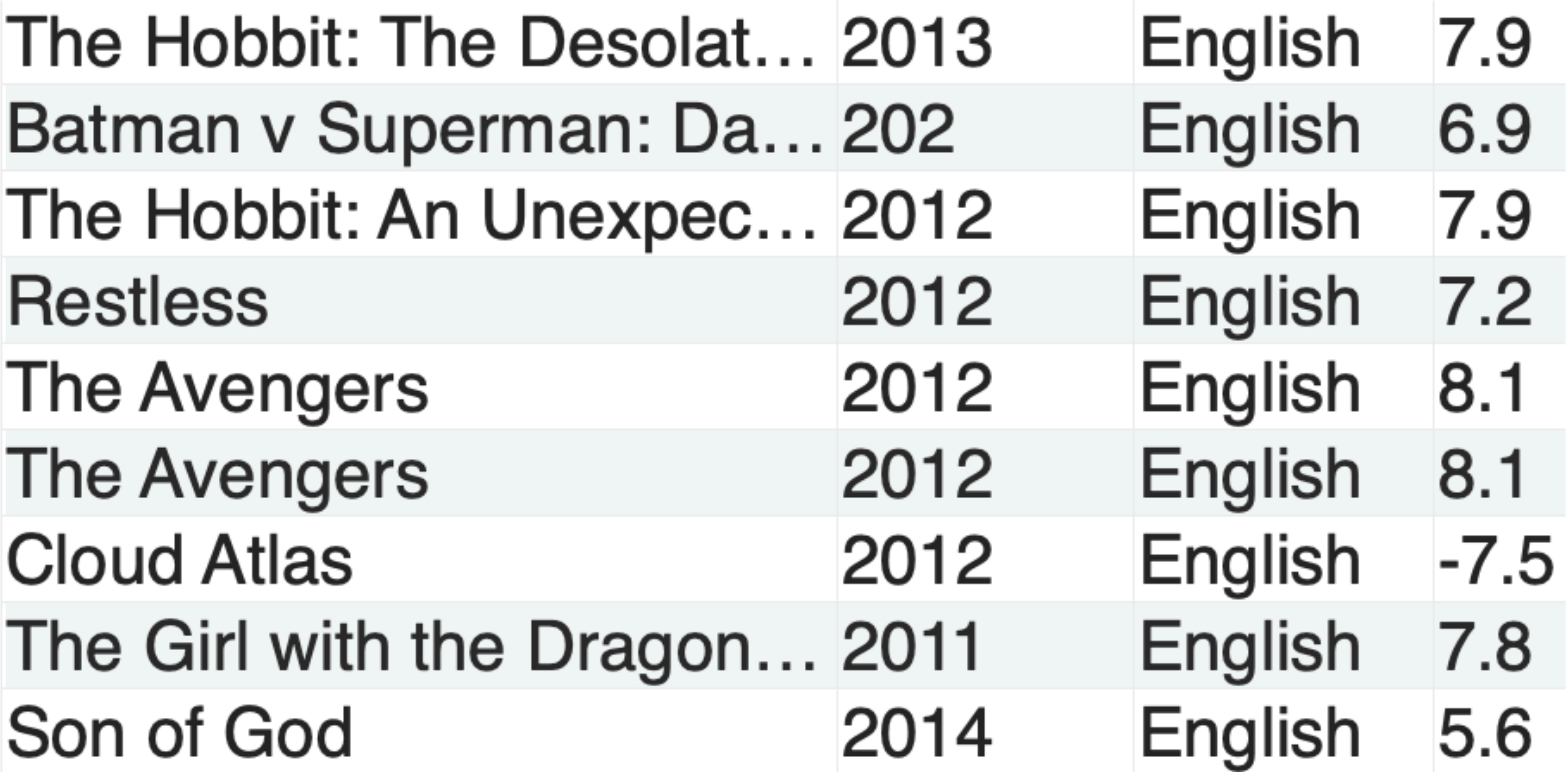
Constraint motives

Data integrity

- What if a value:
  - Has to be unique
  - References a non-existent field
  - Cannot be null
  - Has a default value?
  - Has domain constraints?
    - E.g a song cannot have a negative length
    - An employee cannot receive negative salary

Why does this dataset have low integrity?

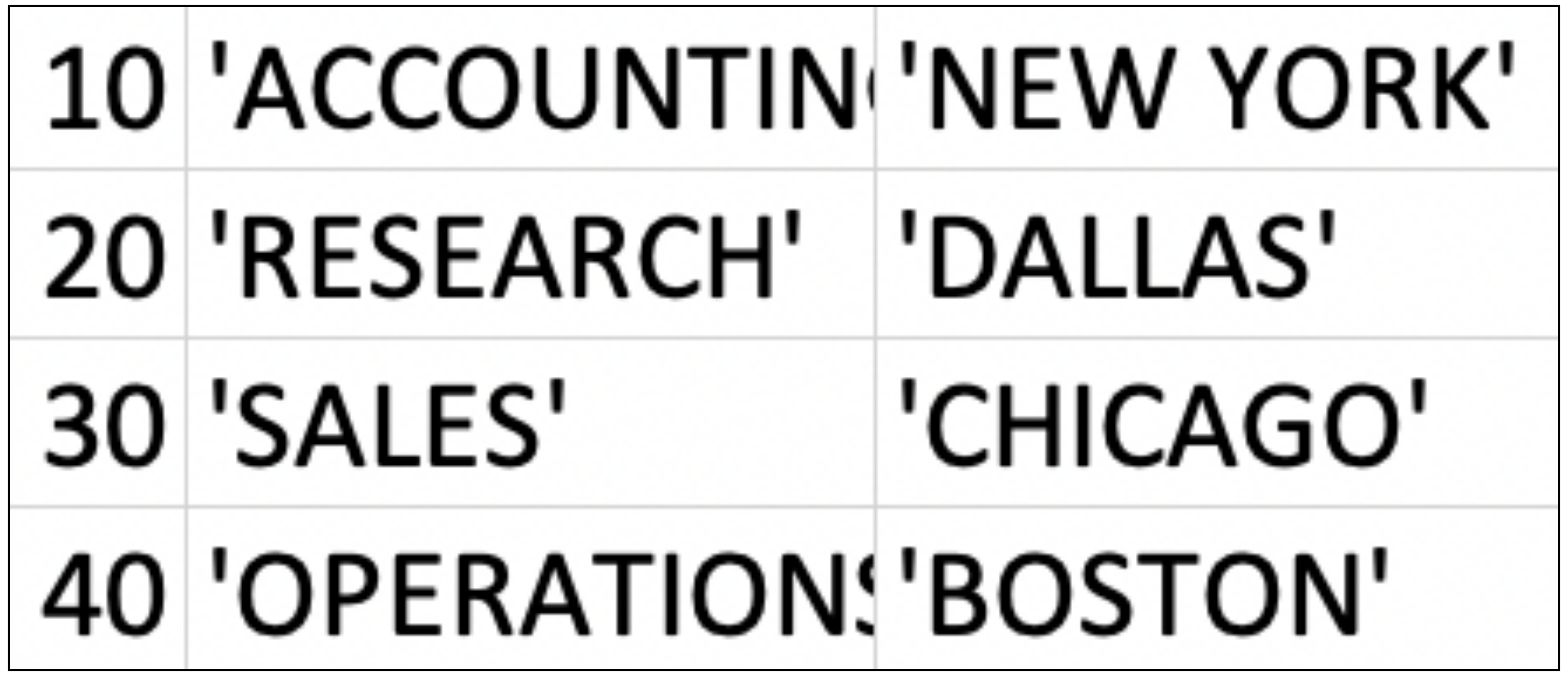
Constraints



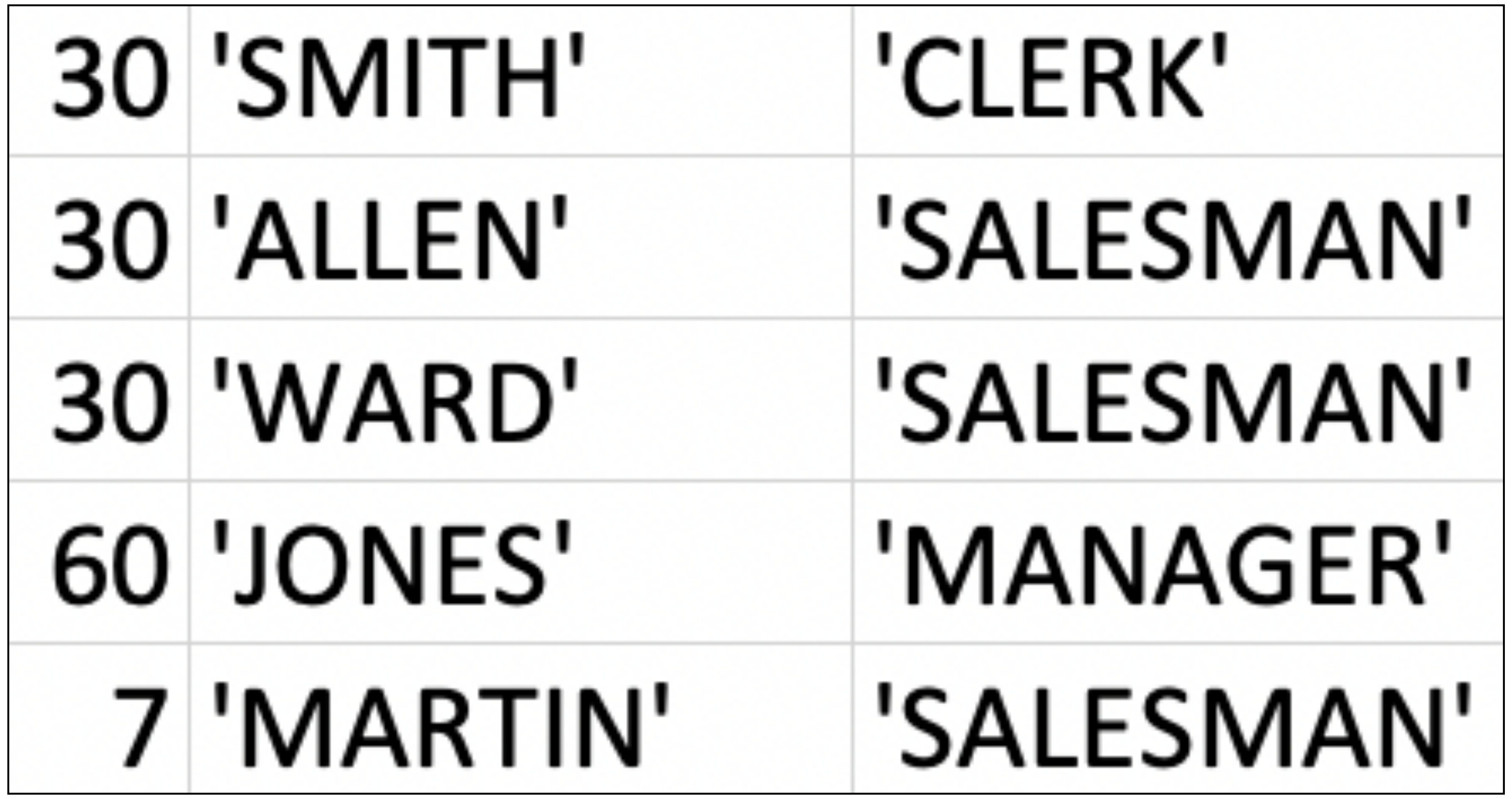
The Hobbit: The Desolat…	2013	English	7.9
Batman v Superman: Da…	202	English	6.9
The Hobbit: An Unexpec…	2012	English	7.9
Restless	2012	English	7.2
The Avengers	2012	English	8.1
The Avengers	2012	English	8.1
Cloud Atlas	2012	English	-7.5
The Girl with the Dragon…	2011	English	7.8
Son of God	2014	English	5.6

Why does this dataset have low integrity?

Constraints



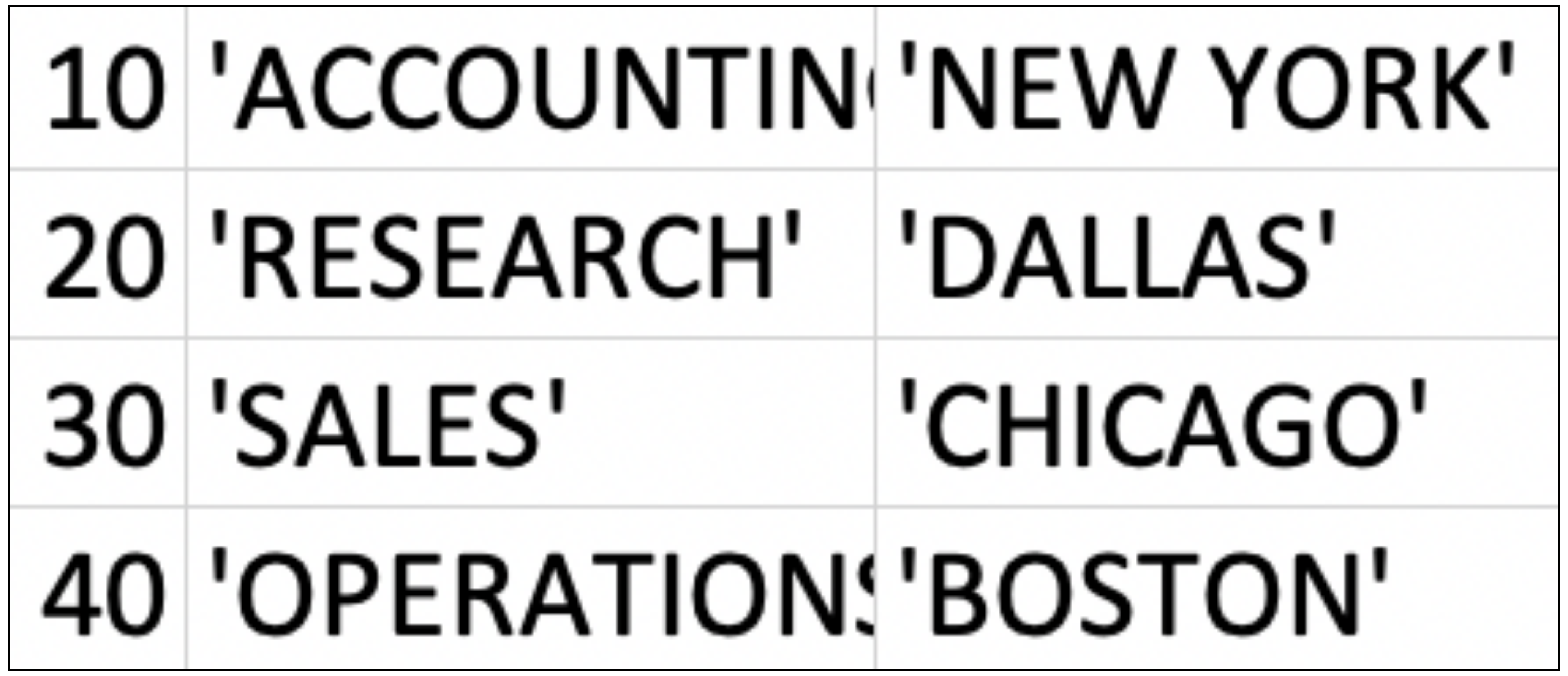
10	'ACCOUNTIN	'NEW YORK'
20	'RESEARCH'	'DALLAS'
30	'SALES'	'CHICAGO'
40	'OPERATIONS	'BOSTON'



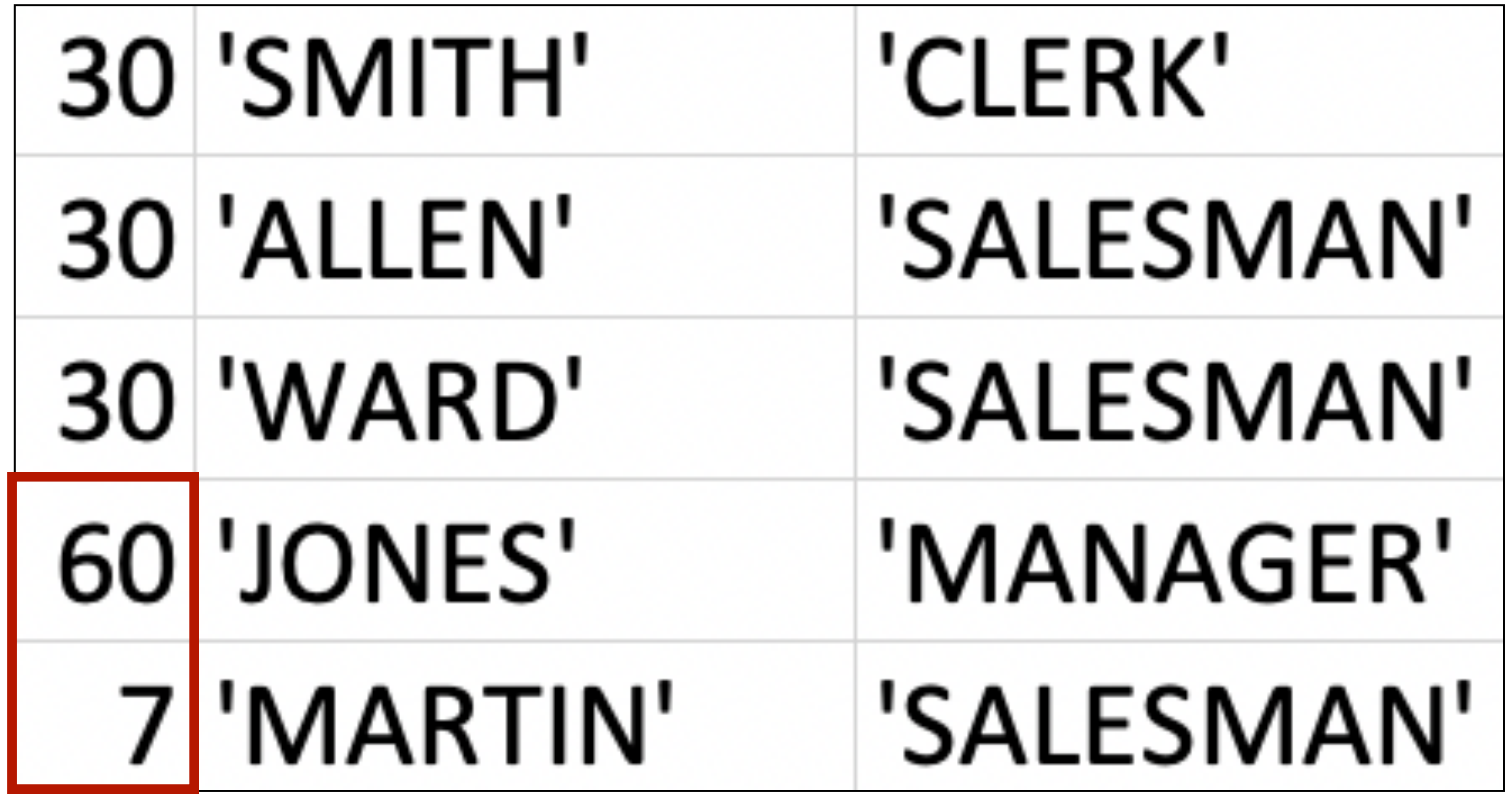
30	'SMITH'	'CLERK'
30	'ALLEN'	'SALESMAN'
30	'WARD'	'SALESMAN'
60	'JONES'	'MANAGER'
7	'MARTIN'	'SALESMAN'

Foreign key references non-existing values

Low Accuracy



10	'ACCOUNTIN	'NEW YORK'
20	'RESEARCH'	'DALLAS'
30	'SALES'	'CHICAGO'
40	'OPERATION	'BOSTON'



30	'SMITH'	'CLERK'
30	'ALLEN'	'SALESMAN'
30	'WARD'	'SALESMAN'
60	'JONES'	'MANAGER'
7	'MARTIN'	'SALESMAN'

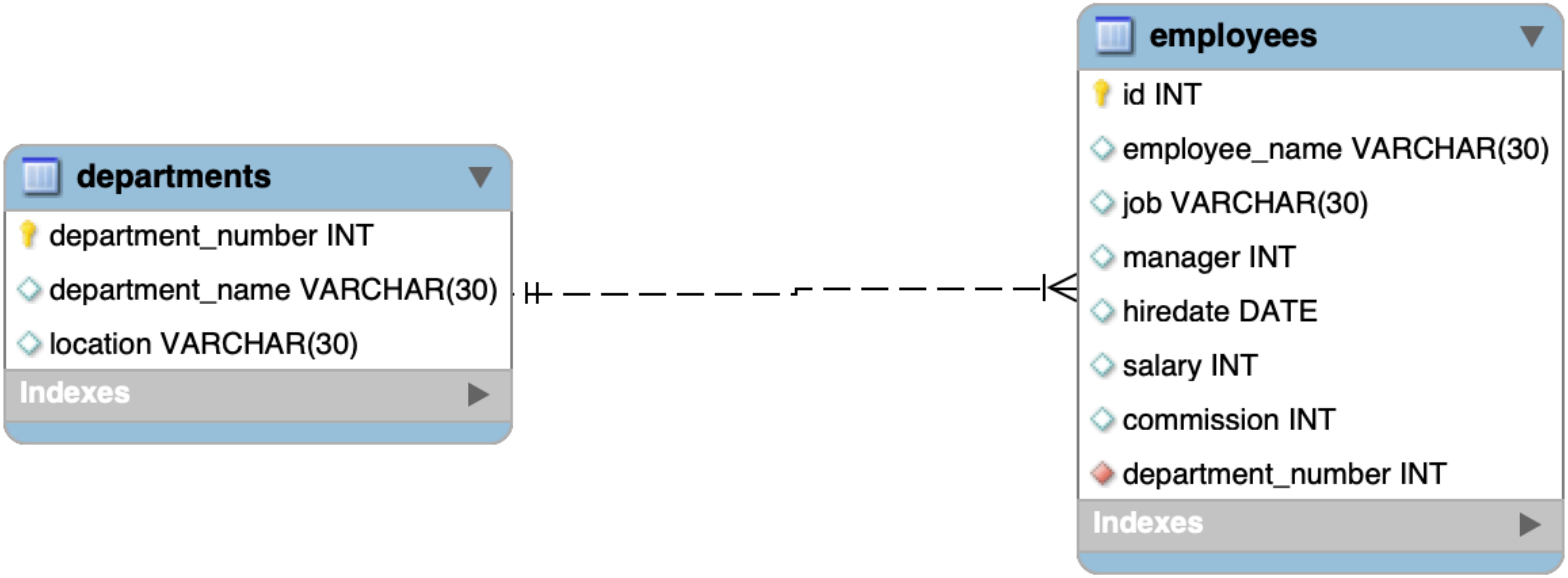
Database design

Constraints

- PRIMARY KEY
    - Unique, NOT NULL, indexed
- NOT NULL
    - Value in column cannot be NULL
- AUTO_INCREMENT
    - When value is inserted, it receives a unique value number n+1 where n is number of rows
- DEFAULT
    - A default value e.g: True, NULL, “Default”

```sql
CREATE TABLE employees(
    id INTEGER,
    employee_name VARCHAR(30),
    job VARCHAR(30),
    manager INTEGER,
    hiredate DATE,
    salary INTEGER,
    commission INTEGER,
    department_number INTEGER,
    PRIMARY KEY (id),
    FOREIGN KEY (department_number) REFERENCES departments(department_number)
);
```

```
CREATE TABLE employees(
    id INTEGER,
    employee_name VARCHAR(30),
    job VARCHAR(30),
    manager INTEGER,
    hiredate DATE,
    salary INTEGER,
    commission INTEGER,
    department_number INTEGER,
    PRIMARY KEY (id),
    FOREIGN KEY (department_number) REFERENCES departments(department_number)
);
```



departments

department_number INT

department_name VARCHAR(30)

location VARCHAR(30)

Indexes

employees

id INT

employee_name VARCHAR(30)

job VARCHAR(30)

manager INT

hiredate DATE

salary INT

commission INT

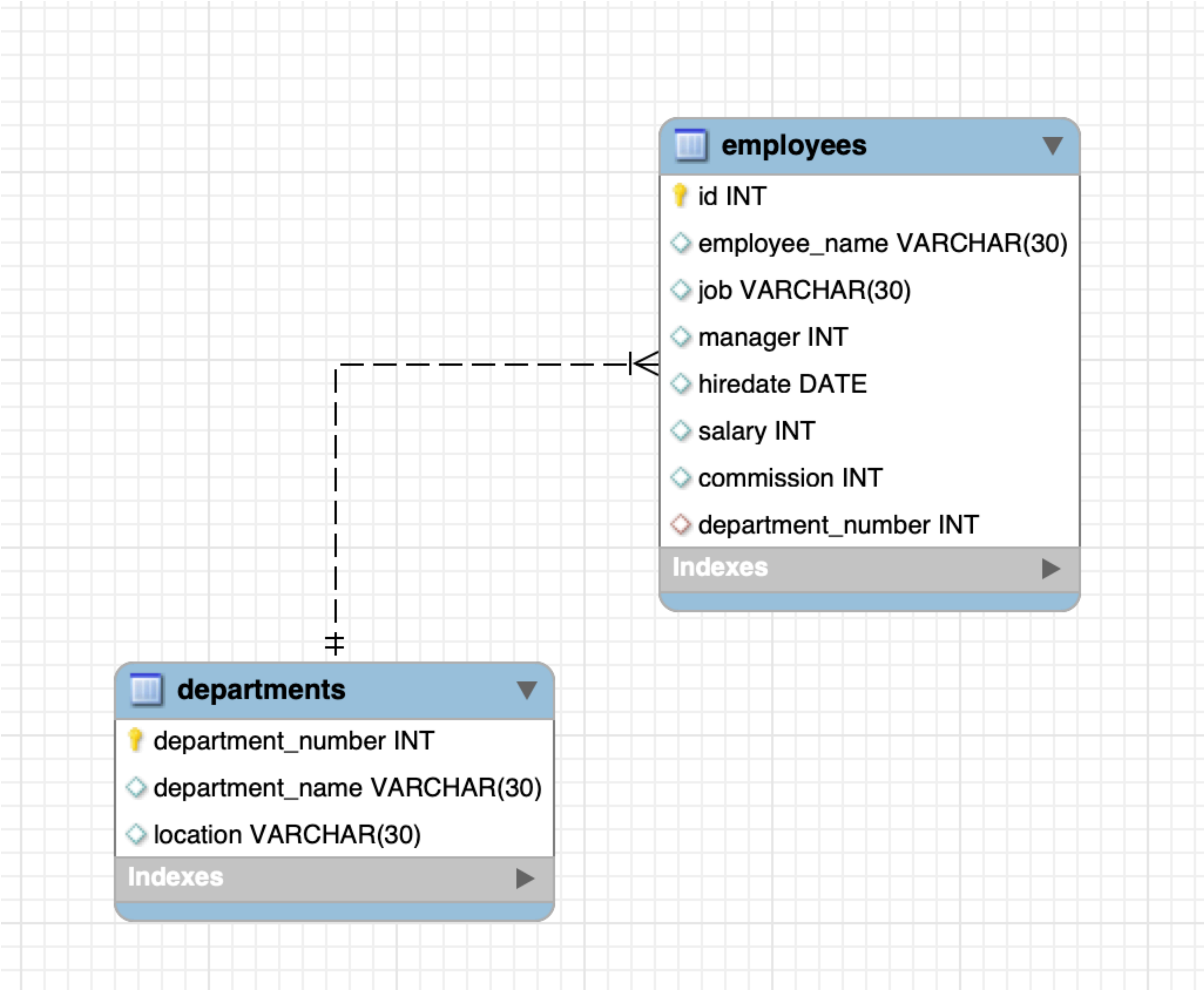
department_number INT

Indexes

Foreign Key Constraints

- What problems can arise if:
  - a department is deleted?
  - a department changes its id?

Referential Integrity



employees

id INT
employee_name VARCHAR(30)
job VARCHAR(30)
manager INT
hiredate DATE
salary INT
commission INT
department_number INT
Indexes

departments

department_number INT
department_name VARCHAR(30)
location VARCHAR(30)
Indexes

ON UPDATE or ON DELETE

Foreign Key Constraints

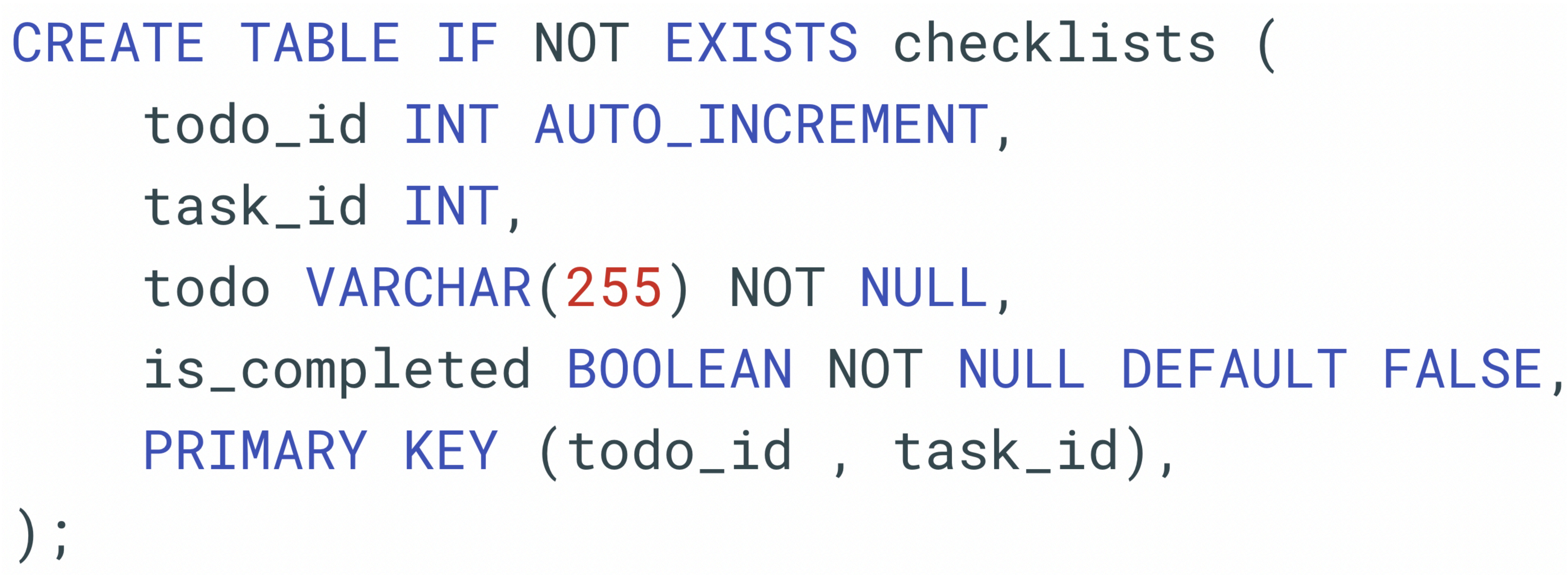
- RESTRICT - not allowed to change referenced rows (DEFAULT)
- CASCADE - Changes will cascade to referencing tables
- SET NULL - Referencing tables will change their values to NULL
- NO ACTION - No change will occur
- SET DEFAULT - Will set referencing tables to their default value (requires a default)

Cascading changes vs. Restrict

Example:
Employees Departments DELETE constraint

```sql
CREATE TABLE DEPT (
    DEPTNO INTEGER,
    DNAME VARCHAR(30),
    LOC VARCHAR(30),
    PRIMARY KEY (DEPTNO)
);
```

What can you tell me about this table?



```sql
CREATE TABLE IF NOT EXISTS checklists (
    todo_id INT AUTO_INCREMENT,
    task_id INT,
    todo VARCHAR(255) NOT NULL,
    is_completed BOOLEAN NOT NULL DEFAULT FALSE,
    PRIMARY KEY (todo_id , task_id),
);
```