RELATIONAL DATABASES

OBJECTIVES

- LEARN ABOUT 'CONSTRAINTS' AS THEY RELATE TO DATA INTEGRITY.
- KNOW WHEN IT IS POSSIBLE TO DEFINE A CONSTRAINT AT THE COLUMN LEVEL AND WHEN IT IS POSSIBLE AT THE TABLE LEVEL.
- KNOW WHAT DATA INTEGRITY RULES ARE ENFORCED BY NOT NULL AND UNIQUE CONSTRAINTS.

CONSTRAINTS

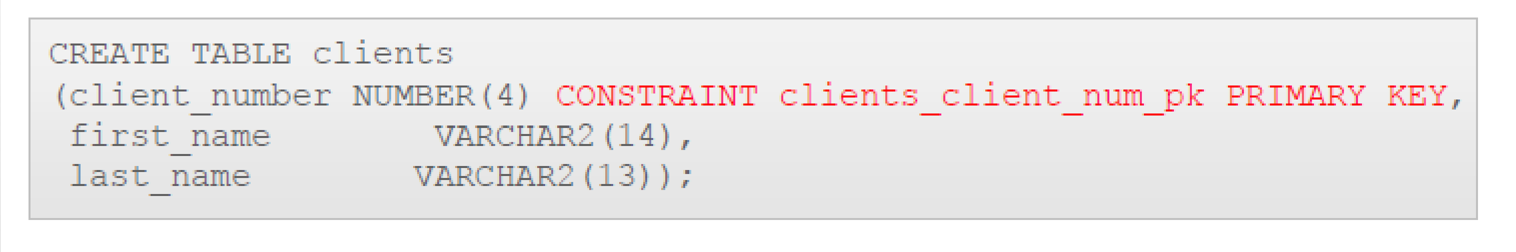
- A CONSTRAINT IS A DATABASE RULE.
- ALL CONSTRAINT DEFINITIONS ARE STORED IN THE DATA DICTIONARY.
- CONSTRAINTS PREVENT THE DELETION OF A TABLE IF THERE ARE DEPENDENCIES FROM OTHER TABLES.
- CONSTRAINTS ENFORCE RULES ON THE DATA WHENEVER A ROW IS INSERTED, UPDATED, OR DELETED FROM A TABLE.
- CONSTRAINTS ARE IMPORTANT AND SO IS NAMING THEM APPROPRIATELY.

CONSTRAINTS

- WE CAN DEFINE OUR CONSTRAINTS IN OUR CREATE TABLE STATEMENT.
- THERE ARE TWO PLACE WE CAN DEFINE CONSTRAINTS IN THE CREATE TABLE STATEMENT,
    - AT THE COLUMN LEVEL NEXT TO THE NAME AND THE DATA TYPE
    - AT THE TABLE LEVEL AFTER ALL THE COLUMN NAMES ARE LISTED.
- THE COLUMN LEVEL REFERS TO WHERE THE COLUMNS ARE DEFINED.
- THE TABLE LEVEL REFERS TO THE LAST LINE IN THE STATEMENT BELOW THE LIST OF COLUMNS.

COLUMN LEVEL CONSTRAINTS

- A COLUMN LEVEL CONSTRAINT REFERENCES A SINGLE COLUMN.
- IT MUST BE DEFINED IN THE CREATE TABLE STATEMENT AS PART OF THE COLUMN DEFINITION.



```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
 first_name        VARCHAR2(14),
 last_name        VARCHAR2(13));
```

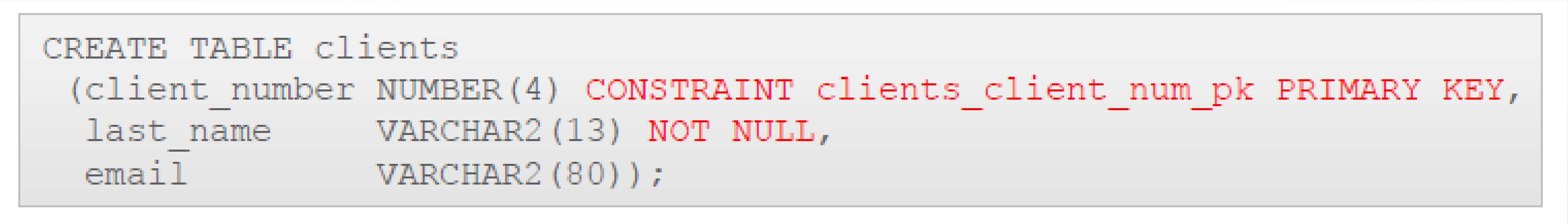
- THE NAME OF THE CONSTRAINT IS CLIENTS_CLIENT_NUM_PK.
- IT ENFORCES THE BUSINESS RULE THAT THE CLIENT_NUMBER IS THE PRIMARY KEY OF THE CLIENTS TABLE.

NAMING CONSTRAINTS

- EVERY CONSTRAINT IN THE DATABASE HAS A NAME. WHEN A CONSTRAINT IS CREATED IT DOES NOT HAVE TO BE GIVEN A NAME BY THE AUTHOR OF THE CREATE TABLE STATEMENT, IN WHICH CASE THE SYSTEM GIVES THE CONSTRAINT A NAME SUCH AS SYS_C0058534.
- A NAMING CONVENTION CAN BE THE COMBINATION OF THE TABLE NAME, A COLUMN NAME AND THE TYPE OF CONSTRAINT.
  - PK FOR PRIMARY KEY, NN FOR NOT NULL, UK FOR UNIQUE KEY, FK FOR FOREIGN KEY.
- IF THE RESERVED WORD CONSTRAINT IS USED IN THE CREATE TABLE DEFINITION, YOU MUST GIVE THE CONSTRAINT A NAME (MAX 30 CHARS).
- IT IS BEST TO NAME CONSTRAINTS AS SYSTEM LEVEL NAMES ARE NOT EASY TO INTERPRET.

NAMING CONSTRAINTS

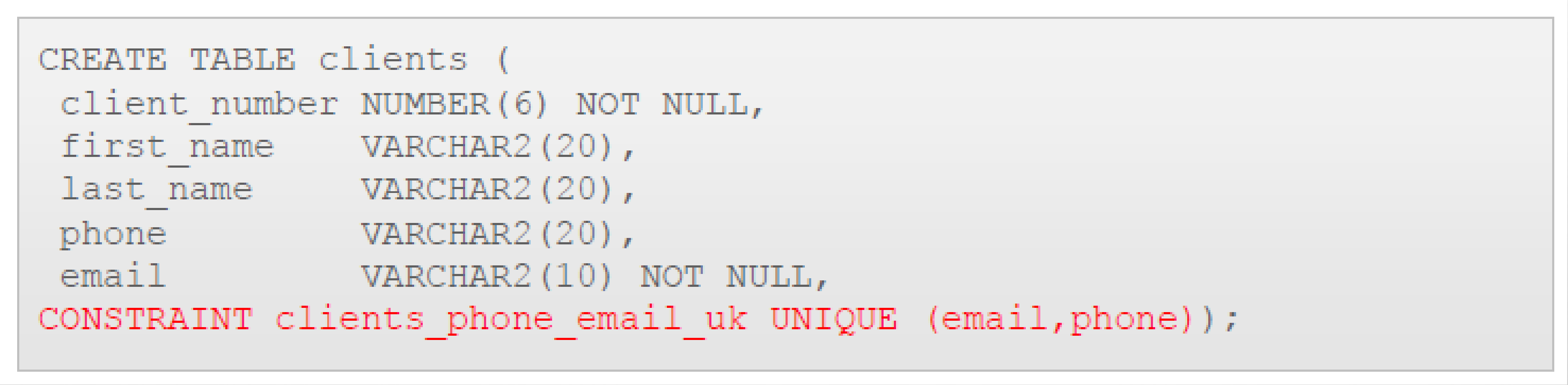
- HERE IS AN EXAMPLE:



```
CREATE TABLE clients
 (client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
  last_name     VARCHAR2(13) NOT NULL,
  email         VARCHAR2(80));
```

TABLE LEVEL CONSTRAINTS

- TABLE LEVEL CONSTRAINTS ARE LISTED SEPARATELY FROM THE COLUMN DEFINITIONS IN THE CREATE TABLE STATEMENT AFTER ALL COLUMNS ARE DEFINED.



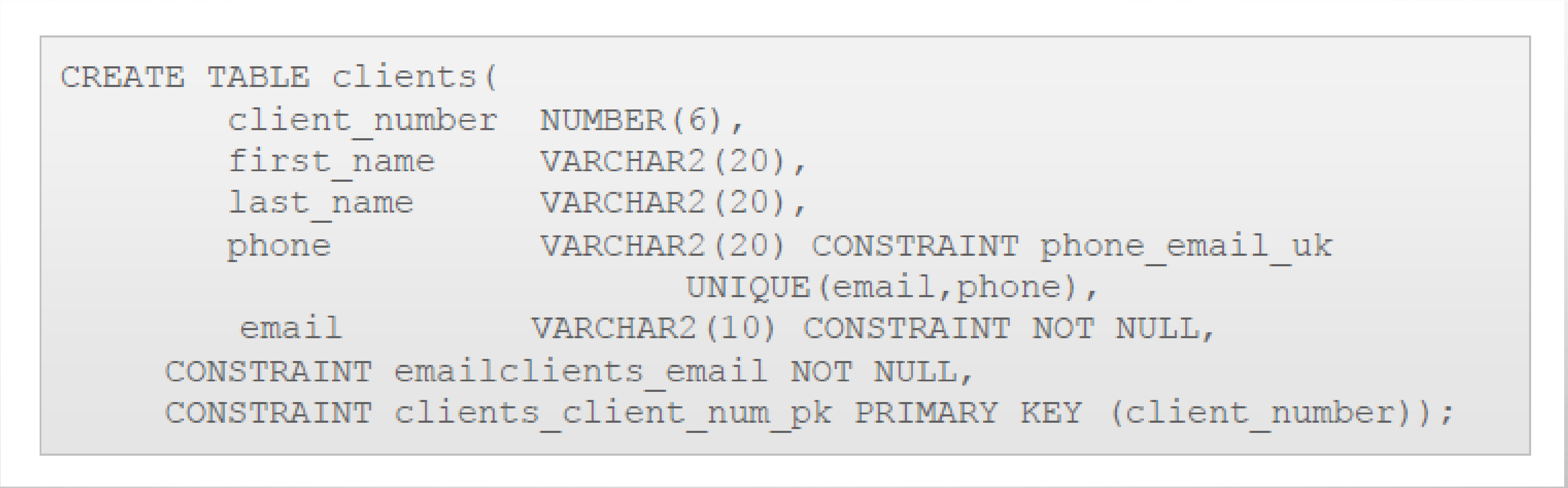
```
CREATE TABLE clients (
 client_number NUMBER(6) NOT NULL,
 first_name    VARCHAR2(20),
 last_name     VARCHAR2(20),
 phone         VARCHAR2(20),
 email         VARCHAR2(10) NOT NULL,
CONSTRAINT clients_phone_email_uk UNIQUE (email,phone));
```

- YOU MUST INCLUDE THE COLUMN(S) NAME FOR WHICH THE CONSTRAINT IS BEING DEFINED.

RULES FOR CONSTRAINTS

- CONSTRAINTS THAT REFER TO MORE THAN ONE COLUMN MUST BE DEFINED AT THE TABLE LEVEL.
- THE NOT NULL CONSTRAINT CAN BE DEFINED ONLY AT THE COLUMN LEVEL.
- UNIQUE, PRIMARY, FOREIGN KEY, AND CHECK CONSTRAINTS CAN BE DEFINED AT EITHER LEVEL.
- IF THE KEYWORD CONSTRAINT IS USED IN THE CREATE TABLE STATEMENT, YOU MUST PROVIDE A NAME FOR THE CONSTRAINT.

CONSTRAINTS VIOLATIONS



```
CREATE TABLE clients(
        client_number  NUMBER(6),
        first_name     VARCHAR2(20),
        last_name      VARCHAR2(20),
        phone          VARCHAR2(20) CONSTRAINT phone_email_uk
                              UNIQUE(email,phone),
         email        VARCHAR2(10) CONSTRAINT NOT NULL,
     CONSTRAINT emailclients_email NOT NULL,
     CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number));
```

CONSTRAINT TYPES

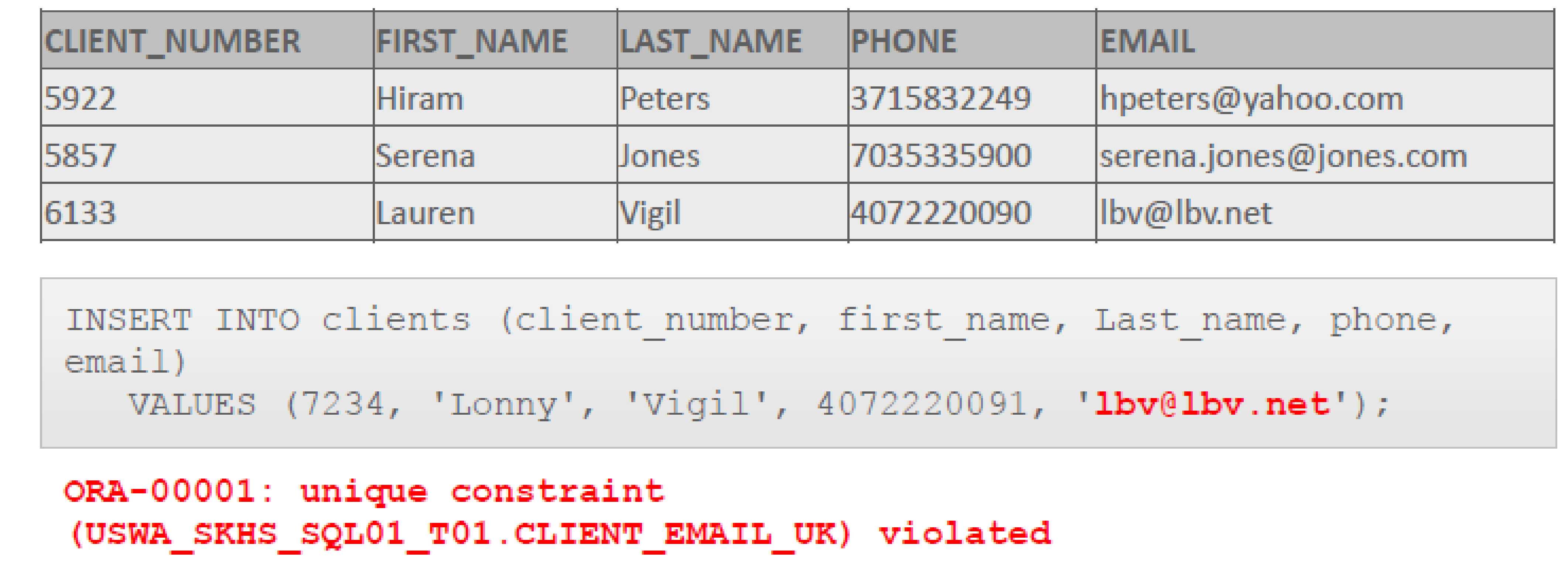
- FIVE TYPES OF CONSTRAINTS:
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

NOT NULL & UNIQUE CONSTRAINTS

- A COLUMN DEFINED WITH A NOT NULL CONSTRAINT REQUIRES THAT FOR EVERY ROW ENTERED INTO THE TABLE, A VALUE MUST EXIST FOR THAT COLUMN.
- IT IS CUSTOMARY TO NAME A NOT NULL CONSTRAINT USING _NN

- A UNIQUE CONSTRAINT REQUIRES THAT EVERY VALUE IN A COLUMN OR SET OF COLUMNS (COMPOSITE) BE UNIQUE; NO TWO ROWS CAN HAVE THE SAME VALUES.
- IT IS CUSTOMARY TO NAME A UNIQUE CONSTRAINT USING _UK
- TO DEFINE A COMPOSITE UNIQUE CONSTRAINT YOU MUST DO IT AT THE TABLE LEVEL
- UNIQUE KEY CONSTRAINT ALLOWS NULL'S UNLESS THE COLUMN ALSO HAS A NOT NULL CONSTRAINT.

CONSTRAINTS

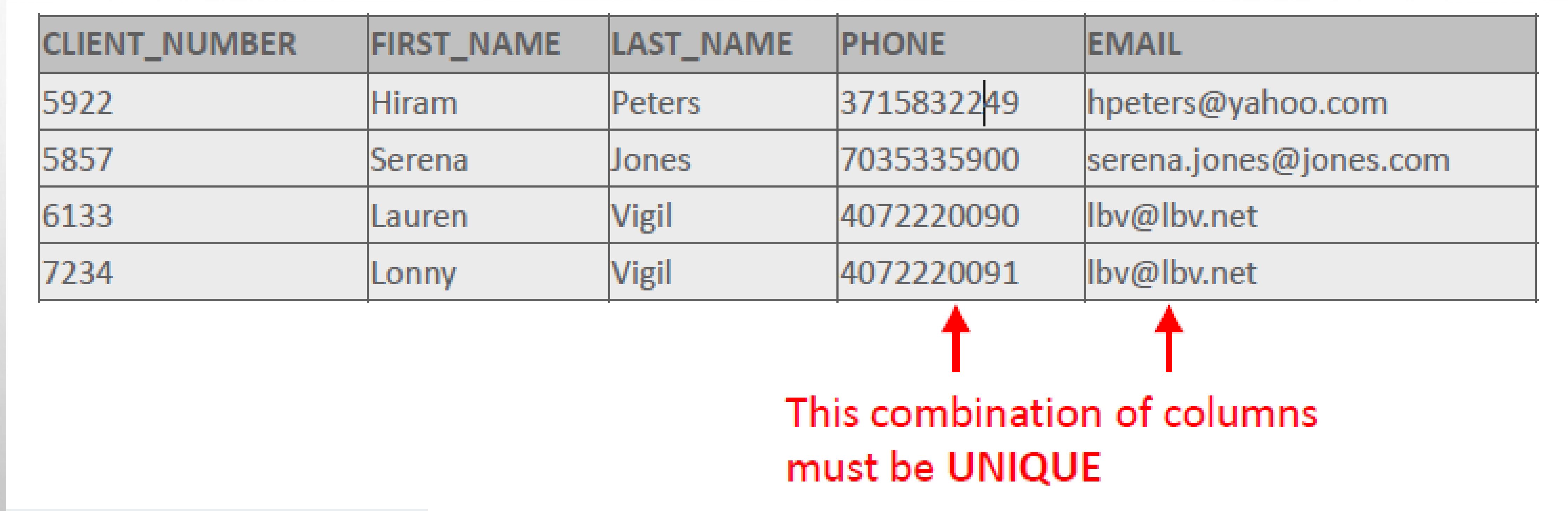


CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena.jones@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net

```
INSERT INTO clients (client_number, first_name, Last_name, phone,
email)
   VALUES (7234, 'Lonny', 'Vigil', 4072220091, 'lbv@lbv.net');
```

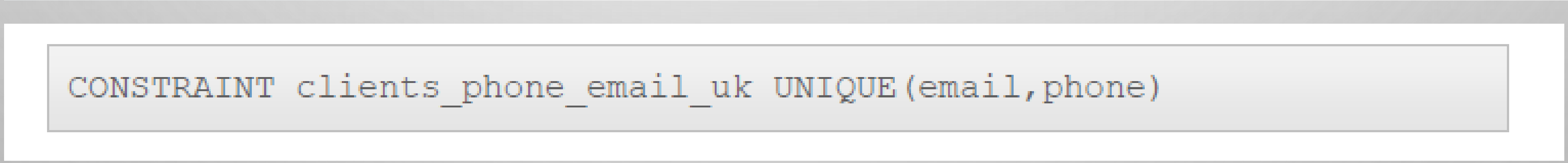
```
ORA-00001: unique constraint
(USWA_SKHS_SQL01_T01.CLIENT_EMAIL_UK) violated
```

CONSTRAINTS



CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena.jones@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net
7234	Lonny	Vigil	4072220091	lbv@lbv.net

This combination of columns must be **UNIQUE**



```
CONSTRAINT clients_phone_email_uk UNIQUE(email,phone)
```