Project Step #4: Data Manipulation Language (DML), 20 Queries, and Consolidated Final Report Submission

An Nguyen

University of Maryland Global Campus

DBST 651: Relational Database System

Professor Mary McDonald

August 9, 2023

TABLE OF CONTENTS

INTRODUCTION 3

OVERVIEW 3

ASSUMPTIONS 3

DESIGN DECISIONS 3

STATEMENT OF WORK 4

REQUIREMENTS DEFINITION DOCUMENT 6

ENTITY RELATIONSHIP DIAGRAM (ERD) 8

ENTITY AND ATTRIBUTE DESCRIPTION 9

DDL SOURCE CODE EMBEDDED 11

DML SOURCE CODE EMBEDDED AND OUTPUT 24

QUERY SOURCE CODE EMBEDDED AND OUTPUT 32

Introduction

This technical report explains the implemented immigration system in a governmental setting that processes and keeps track of immigrant applications. The examples used in this document aim to present results that facilitate record-keeping and tracking. The DBMS system seeks to improve retention time and track delay factors in processing applications. It provides insight into each social worker's workload and helps management make decisions that reassess each social worker's capabilities. In addition, it keeps a statistic about the immigrant's origin for future study.

Overview

The database will receive information directly from the application server with regard to the applicant's information, keeping track of workers, and receiving modifications from workers who are authorized to access the applicant's information. The database allows workers to cross-reference all of the documents submitted by the applicant. It has a function that allows management to view workloads and reallocate them where they make the most sense. The application receives input from both the worker and the applicant.

Assumptions

A worker can work on multiple applications at once, however, each application can only be assigned to a social worker. This problem could lead to work overload, so amount of application assign to each worker should be revisit for assessment.

Design Decisions

The design of the database consists of five entities established through the entity-relationship diagram. The Person entity represents the applicant (input). The Worker entity represents the user. The Application entity represents the user input data used to track progress. The Document entity is used as a placeholder for the applicant's other information. The Origin entity is used to collect data for future studies based on its statistics.

Statement of Work

1. Overview

Due to mass migration from Central and South America into North America, crime rates have spiked. The country needs a database system that will centralize immigrant demographics, criminal and health background, skills, education, employment history, and reasons for migration. The system will grant permission to enter the country based on ability to contribute and deny permission to criminals such as the drug cartel, human traffickers, convicts, or people who will abuse the welfare system. The database project aims to promote successful integration through job placement, educational assistance programs, and support services for returning to your home country.

2. Purpose and Objectives:

2.1 Database project objectives (benefits)

a. The database will be developed on a scalable database system that can support large volumes of immigration data. It will capture information about the immigrants, such as their origin country, educational background, employment history, language skills, health background, such as previous or current signs of substance abuse, and personal details, including physical characteristics.

b. The information will be embedded into an identification code similar to a social security number that will prompt the individual to renew biannually for follow-up and update necessary information such as residential address or employment.

c. Immigration agents and government entities can access the information using the identification code to update the immigrant's information and match the immigrant with suitable employment opportunities based on skills, qualifications, and preferences.

d. The system will generate reports and track the immigrant's progress in assimilation, including access to education, housing, healthcare, employment, and contributions to society.

e. The generated report and analytics will be used to assess the eligibility of the immigrant for renewal of their issued identification or rehoming them in their origin country. It will be used to track and monitor immigrants through an identification code until they become full-fledged citizens or until they are rehomed in their origin country.

2.2 Database project Scope

This database is designed to be implemented in the immigration office and city and state social service agencies offices. The system will be proprietary software only for government usage. Data manipulation and access are granted to trained and authorized personnel working directly with the individual files or cases.

In-Scope Work

Entity-relationship model

DDL Scripts

DML Scripts

SQL Query Scripts

Technical Report

Out-of-Scope Work

Acquiring feedback to improve Database flow

Implementation database

3. Hardware and Software

3.1 Hardware

The server will be hosted on the Oracle database cloud service. Onsite computers with the latest updates of anti-virus and anti-malware and the latest version of Windows should be able to access the server from applications on the desktop.

3.2 Software

The following software will be used in this project:

- Oracle Database 19c for user to access information or add information and for regular maintenance of the data such as update or necessary modification

- SQL Developer 23.1 Version 23.1.0.097.1607 - April 19, 2023 for database engineer to make changes to database structures
- ER Assistant Version 2.10 for schema diagram drafting

4. SQL Usage and Style

SQL usage and style will follow *Oracle Documentation Standard by Oracle*, available at https://docs.oracle.com/pdf/E23911_09.pdf.

Additional guide for developer adapted from *Oracle SQL and PL/SQL Optimization for Developers by Hellström* (2023), available at https://oracle.readthedocs.io/en/latest/index.html

a. Data definition language (DDL) will be use to build or modify structures of the table and objects in the database.
b. Data manipulation language (DML) will be use to work with data in the tables
c. Naming Convention: Column name must contain A to Z and underscore (_) must be use in place of space or omit space, no number should be use in column naming, names are not case sensitive i.e., FIRST_NAME, Last_name, Country_of_Origin etc.…
d. Date Format: DD-MMM-YY i.e., 10Jun21
e. Table rules: Each table must have a primary key or a foreign key, description of table

Requirements Definition Document

Relationship: "works" between PERSON and APPLICATION

Cardinality: 1:1 between PERSON and APPLICATION

Business Rule: Each person can only submit one application.

Relationship: "works" between PERSON and DOCUMENT

Cardinality: 1:M between PERSON and DOCUMENT

Business Rule: A person can have multiple documents, such as a birth certificate, health report, university degree, etc.; however, those documents belong to one specific person.

Relationship: "works" between PERSON and CASE_WORKER

Cardinality: M:M between PERSON and CASE_WORKER

Business Rule: A person can have more than one case worker working on their files depending on the stages of the application, and a case worker might be working with more than one person, hence more than one application.

Relationship: “works” between CASE_WORKER and APPLICATION

Cardinality: 1:M between CASE_WORKER and APPLICATION

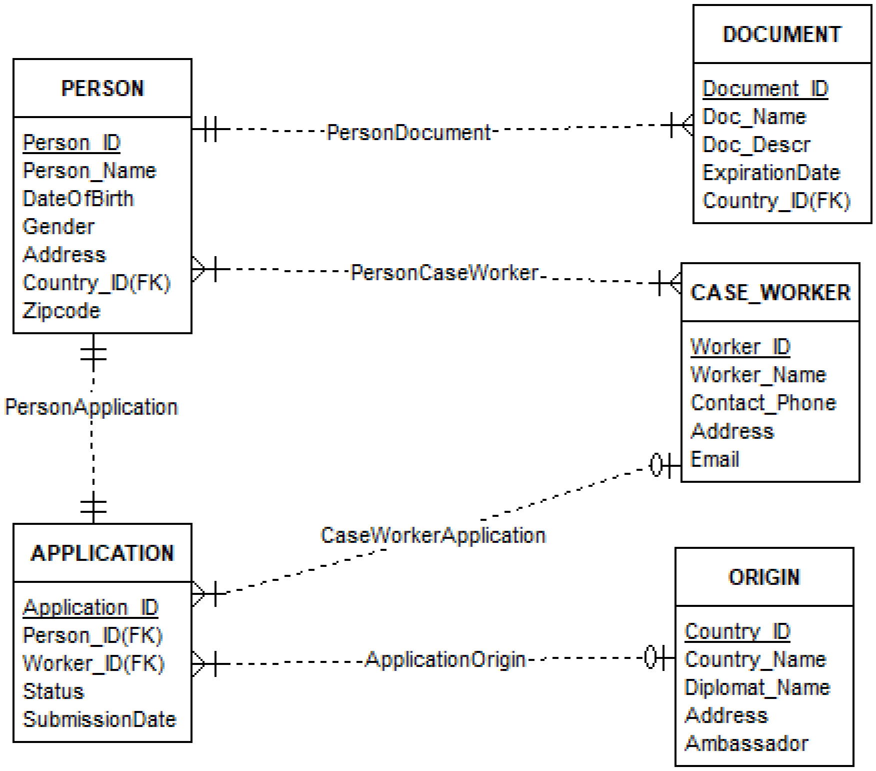
Business Rule: A case worker might be working on one or many applications at any point in time.

Relationship: “works” between ORIGIN and APPLICATION

Cardinality: 1:M between ORIGIN and APPLICATION

Business Rule: Many applications can come from one country but one applicant (person) cannot have citizenship of many countries. Each country might have multiple applicants (people) applying for immigration. Each application is strictly associated with one country.

Entity Relationship Diagram (ERD)



Entity and Attribute Description

Entity Name: Person

Parent Entity: N/A
Primary Key: Person_ID
Foreign Key: N/A
Attribute Name: Person_ID.
Attribute Description: Identification number assigned to each person upon entry into the country.
Attribute Name: Person_Name
Attribute Description: First and last name of the person.
Attribute Name: DateOfBirth
Attribute Description: The person date of birth, verifying against birth certificate.
Attribute Name: Gender
Attribute Description: The person 's biological gender.
Attribute Name: Address
Attribute Description: The person's residential address or temporary housing address.
Attribute Name: Country_ID
Attribute Description: Referencing country id from Origin as foreign key
Attribute Name: Zipcode
Attribute Description: zipcode of the address

Entity Name: Application

Parent Entity: Person, Case_Worker
Primary Key: Application_ID
Foreign Key: Person_ID, Worker_ID
Attribute Name: Application_ID.
Attribute Description: The identification number of the application assigned on application submission date.
Attribute Name: Person_ID
Attribute Description: A foreign key is used to tie each person to each application.
Attribute Name: Worker_ID
Attribute Description: A foreign key is used to tie each application to the current case worker.
Attribute Name: Status
Attribute Description: Showing current status of the application such as pending, review, approved, rejected.
Attribute Name: SubmissionDate
Attribute Description: Showing the date of application submission.

Entity Name: Document

Parent Entity: Origin
Primary Key: Document_ID
Foreign Key: Country_ID
Attribute Name: Document_ID.
Attribute Description: The identification number of each submitted document.
Attribute Name: Doc_Name
Attribute Description: The submitted documentation name.
Attribute Name: Doc_Descr
Attribute Description: A brief description of each submitted document.
Attribute Name: ExpirationDate
Attribute Description: Submitted document expiration date, it can be NULL for documentation with no expiration date such as birth certification or university degree.
Attribute Name: Country_ID
Attribute Description: A foreign key is used to trace the document's issuing country.

Entity Name: Case_Worker

Parent Entity: N/A
Primary Key: Worker_ID
Foreign Key: N/A
Attribute Name: Worker_ID
Attribute Description: The identification number of each case worker.
Attribute Name: Worker_Name
Attribute Description: First and last name of the case worker.
Attribute Name: Contact_Phone
Attribute Description: Business hour contact phone number.
Attribute Name: Address
Attribute Description: Case worker's residential address.
Attribute Name: Email
Attribute Description: Case worker’s business email issued by associated agency.

Entity Name: Origin

Parent Entity: N/A

Primary Key: Country_ID

Foreign Key: N/A

Attribute Name: Country_ID

Attribute Description: The identification number of each country.

Attribute Name: Country_Name

Attribute Description: Name of the country

Attribute Name: Ambassador

Attribute Description: A Boolean value attribute confirms if the origin country has an ambassador with the current country (yes, true, or no, false).

Attribute Name: Address

Attribute Description: Current address of the ambassador.

Attribute Name: Diplomat_Name

Attribute Description: First and last name of the country of origin representative.

DDL Source Code Embedded

```
/*
Nguyen_An
Project Step #4: SQL DDL and DML Script
DBST 651
University of Maryland Global Campus
*/
-- drop database and create if not exist
DROP DATABASE IF EXISTS Immigrationdb;
CREATE DATABASE        IF NOT EXISTS Immigrationdb;
USE Immigrationdb;
-- drop tables
DROP TABLE IF EXISTS Origin;
DROP TABLE IF EXISTS Person;
DROP TABLE IF EXISTS Worker;
DROP TABLE IF EXISTS Application;
DROP TABLE IF EXISTS Document;
--  drop views
DROP VIEW IF EXISTS OriginInfo;
DROP VIEW IF EXISTS PersonInfo;
DROP VIEW IF EXISTS WorkerInfo;
DROP VIEW IF EXISTS ApplicationInfo;
DROP VIEW IF EXISTS DocumentInfo;
-- drop triggers
DROP TRIGGER IF EXISTS TRG_Origin_Insert;
DROP TRIGGER IF EXISTS TRG_Origin_Update;
DROP TRIGGER IF EXISTS TRG_Person_Insert;
DROP TRIGGER IF EXISTS TRG_Person_Update;
DROP TRIGGER IF EXISTS TRG_Worker_Insert;
DROP TRIGGER IF EXISTS TRG_Worker_Update;
DROP TRIGGER IF EXISTS TRG_Application_Insert;
DROP TRIGGER IF EXISTS TRG_Application_Update;
DROP TRIGGER IF EXISTS TRG_Document_Insert;
DROP TRIGGER IF EXISTS TRG_Document_Update;

/*Create tables for the entities*/
CREATE TABLE Origin (
   country_id             INTEGER      NOT NULL AUTO_INCREMENT,
   country_name             VARCHAR(30)   NOT NULL,
   diplomat_name             VARCHAR(30)    NOT NULL,
   address                VARCHAR(512)      NOT NULL,
   ambassador_add                VARCHAR(30)   NOT NULL,
         created_by                        VARCHAR(30),
   date_created                          DATE,
   modified_by                           VARCHAR(30),
   date_modified                         DATE,
   PRIMARY KEY (country_id)
);
CREATE TABLE Person (
   person_id            INTEGER      NOT NULL    AUTO_INCREMENT,
   country_id                        INTEGER      NOT NULL,
   person_name           VARCHAR(50) NOT NULL,
   dateofbirth          DATE       NOT NULL,
   gender                        VARCHAR(10)        NOT NULL,
   address                           VARCHAR(512)       NOT NULL,
   zipcode                                   VARCHAR(10)                    NOT NULL,
   created_by                        VARCHAR(30),
   date_created                      DATE,
   modified_by                       VARCHAR(30),
   date_modified                     DATE,

         PRIMARY KEY (person_id),
   FOREIGN KEY (country_id) REFERENCES Origin (country_id)
);
CREATE TABLE Worker (
   worker_id            INTEGER              NOT NULL AUTO_INCREMENT,
   worker_name            VARCHAR(30)     NOT NULL,
   contact_phone          VARCHAR(22)     NOT NULL,
   address              VARCHAR(512)   NOT NULL,
   email                          VARCHAR(30)       NOT NULL,
created_by                        VARCHAR(30),
   date_created                       DATE,
   modified_by                        VARCHAR(30),
   date_modified                      DATE,

         PRIMARY KEY (worker_id)
);
```

```
CREATE TABLE Application (
   application_id          INTEGER      NOT NULL AUTO_INCREMENT,
   person_id               INTEGER      NOT NULL,
   worker_id               INTEGER        NOT NULL,
   app_status             VARCHAR(30)        NOT NULL,
   submit_date             DATE         NOT NULL,
         created_by                       VARCHAR(30),
   date_created                           DATE,
   modified_by                            VARCHAR(30),
   date_modified                          DATE,

   PRIMARY KEY (application_id),
   FOREIGN KEY (person_id) REFERENCES Person (person_id),
   FOREIGN KEY (worker_id) REFERENCES Worker (worker_id)
);
CREATE TABLE Document (
   document_id             INTEGER      NOT NULL AUTO_INCREMENT,
   application_id                         INTEGER              NOT NULL,
   doc_name                VARCHAR(90)   NOT NULL,
   doc_descr               VARCHAR(512)     NOT NULL,
   expiration_date          DATE                                          ,
   country_id                       INTEGER      NOT NULL,
         created_by                       VARCHAR(30),
   date_created                           DATE,
   modified_by                            VARCHAR(30),
   date_modified                          DATE,

         PRIMARY KEY (document_id),
   FOREIGN KEY (country_id) REFERENCES Origin (country_id),
   FOREIGN KEY (application_id) REFERENCES Application (application_id)
);
```

```
-- country ID Foreign Keys
CREATE INDEX IDX_Origin_country_id_FK ON Origin (country_id);

-- person ID Foreign Keys
CREATE INDEX IDX_Person_person_id_FK ON Person (person_id);

-- worker ID Foreign Keys
CREATE INDEX IDX_Worker_worker_id_FK ON Worker (worker_id);

-- Frequent-queried columns
CREATE INDEX IDX_Person_person_name ON Person (person_name);



/* Create Views */

-- Business purpose: The Origin view will be used to populate the Person information from
their home country.
CREATE OR REPLACE VIEW OriginInfo AS
SELECT country_id, country_name, diplomat_name, address, ambassador_add
FROM Origin;

-- Business purpose: The PersonInfo view will be used primarily for rapidly fetching
information about individual applicant.
CREATE OR REPLACE VIEW PersonInfo AS
SELECT  person_id, person_name, dateofbirth, gender, address
FROM Person;

-- Business purpose: The WorkerInfo view will be used to fetch information about an the
social worker who is working on the applicant's case.
CREATE OR REPLACE VIEW WorkerInfo AS
```

```
SELECT worker_id, worker_name, contact_phone, address, email
FROM Worker;

-- Business purpose: The Application view will be used to populate a list of application
submitted by the person.
CREATE OR REPLACE VIEW ApplicationInfo AS
SELECT application_id, person_id, worker_id, app_status, submit_date
FROM Application;



-- Business purpose: The Origin view will be used to populate the Person information from
their home country.
CREATE OR REPLACE VIEW DocumentInfo AS
SELECT country_id, country_name, diplomat_name, address, ambassador_add
FROM Origin;

DELIMITER //

-- Business purpose: The TRG_Origin_Insert trigger automatically assigns a sequential
country ID to a newly-inserted row in the Origin table.

DELIMITER //

CREATE TRIGGER TRG_Origin_Insert
        BEFORE INSERT ON Origin
        FOR EACH ROW
   BEGIN
                IF NEW.country_id IS NULL THEN
                        SET NEW.country_id := (SELECT COALESCE(MAX(country_id), 0)
+ 1 FROM Origin);
                END IF;
```

```
    IF NEW.created_by IS NULL THEN
                  SET NEW.created_by := CURRENT_USER();
            END IF;
    IF NEW.date_created IS NULL THEN
                  SET NEW.date_created := CURDATE();
            END IF;
END;
//
DELIMITER;
-- Business rule: TRG_Origin_Update is setting the join date to the current system date and assigning appropriate values to the created_by and date_created fields.
-- If the record is being updated, appropriate values are assigned to the modified_by and modified_date fields.

DELIMITER //

CREATE TRIGGER TRG_Origin_Update
      BEFORE UPDATE ON Origin
      FOR EACH ROW
  BEGIN
            IF NEW.modified_by IS NULL THEN
                  SET NEW.modified_by := CURRENT_USER();
            END IF;
    IF NEW.date_modified IS NULL THEN
                  SET NEW.date_modified := NOW();
            END IF;
END;
//

DELIMITER;
```

```
-- Business purpose: The TRG_Person_Insert trigger automatically assigns a sequential person ID to a newly-inserted row in the Person table.
DELIMITER //

CREATE TRIGGER TRG_Person_Insert
    BEFORE INSERT ON Person
    FOR EACH ROW
 BEGIN
        IF NEW.person_id IS NULL THEN
            SET NEW.person_id := (SELECT COALESCE(MAX(person_id), 0) + 1 FROM Person);
        END IF;
   IF NEW.created_by IS NULL THEN
            SET NEW.created_by := CURRENT_USER();
        END IF;
   IF NEW.date_created IS NULL THEN
            SET NEW.date_created := CURDATE();
        END IF;
END;
//
DELIMITER;

-- Business rule: TRG_Person_Update is setting the join date to the current system date and assigning appropriate values to the created_by and date_created fields.
-- If the record is being updated, appropriate values are assigned to the modified_by and modified_date fields.

DELIMITER //

CREATE TRIGGER TRG_Person_Update
    BEFORE UPDATE ON Person
```

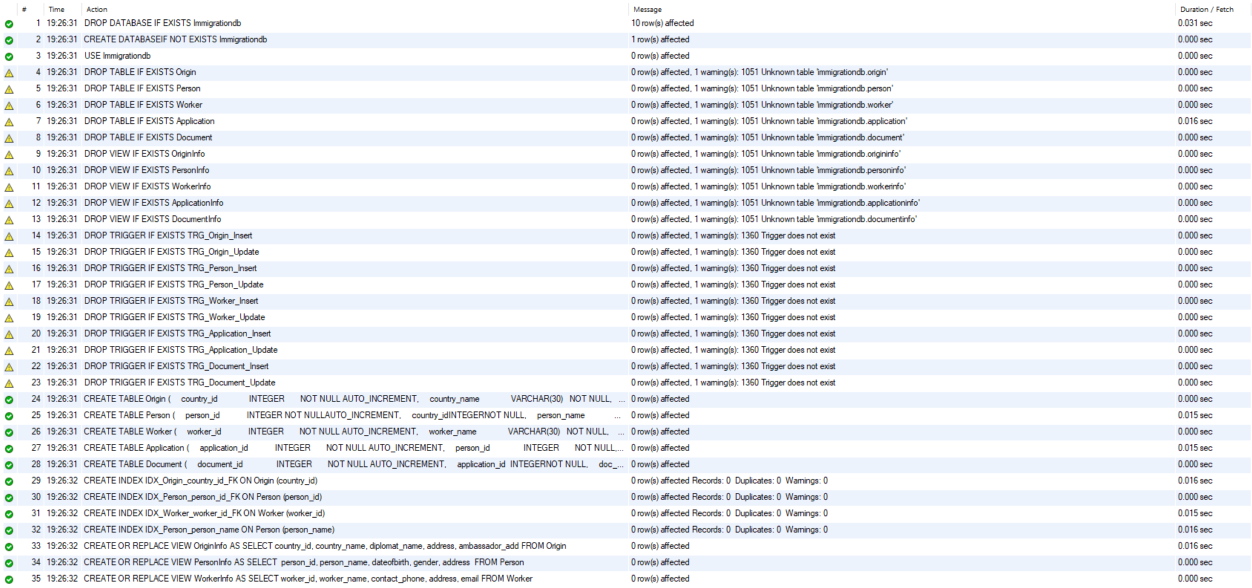
```
      FOR EACH ROW
  BEGIN
            IF NEW.modified_by IS NULL THEN
                  SET NEW.modified_by := CURRENT_USER();
            END IF;
    IF NEW.date_modified IS NULL THEN
                  SET NEW.date_modified := NOW();
            END IF;
END;
//

DELIMITER;

-- Business purpose: The TRG_Worker_Insert trigger automatically assigns a sequential worker ID to a newly-inserted row in the Worker table.
DELIMITER //

CREATE TRIGGER TRG_Worker_Insert
      BEFORE INSERT ON Worker
      FOR EACH ROW
  BEGIN
            IF NEW.worker_id IS NULL THEN
                  SET NEW.worker_id := (SELECT COALESCE(MAX(person_id), 0) + 1 FROM Worker);
            END IF;
    IF NEW.created_by IS NULL THEN
                  SET NEW.created_by := CURRENT_USER();
            END IF;
    IF NEW.date_created IS NULL THEN
                  SET NEW.date_created := CURDATE();
            END IF;
```

```
END;
//
DELIMITER;

-- Business rule: TRG_Worker_Update is setting the join date to the current system date and assigning appropriate values to the created_by and date_created fields.
-- If the record is being updated, appropriate values are assigned to the modified_by and modified_date fields.

DELIMITER //

CREATE TRIGGER TRG_Worker_Update
            BEFORE UPDATE ON Worker
            FOR EACH ROW
    BEGIN
                        IF NEW.modified_by IS NULL THEN
                                    SET NEW.modified_by := CURRENT_USER();
                        END IF;
        IF NEW.date_modified IS NULL THEN
                                    SET NEW.date_modified := NOW();
                        END IF;
END;
//

DELIMITER;
-- Business purpose: The TRG_Application_Insert trigger automatically assigns a sequential application ID to a newly-inserted row in the Application table.

DELIMITER //

CREATE TRIGGER TRG_Application_Insert
```

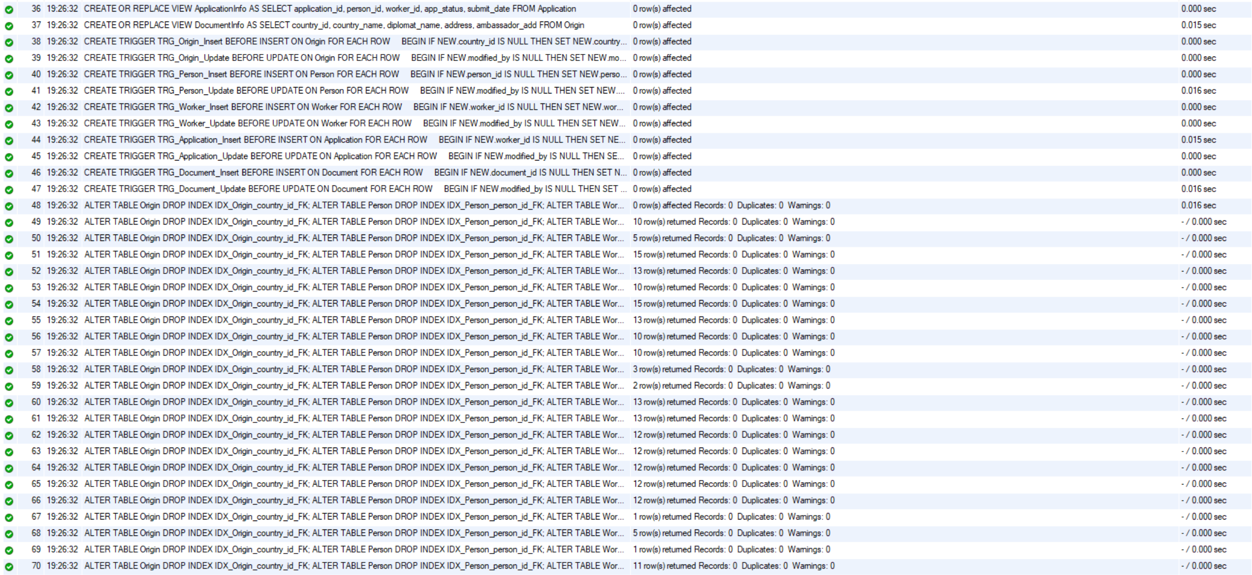
```
            BEFORE INSERT ON Application
            FOR EACH ROW
    BEGIN
                        IF NEW.worker_id IS NULL THEN
                                    SET NEW.application_id := (SELECT COALESCE(MAX(person_id),
0) + 1 FROM Application);
                        END IF;
        IF NEW.created_by IS NULL THEN
                                    SET NEW.created_by := CURRENT_USER();
                        END IF;
        IF NEW.date_created IS NULL THEN
                                    SET NEW.date_created := CURDATE();
                        END IF;
END;
//
DELIMITER;
-- Business rule: TRG_Application_Update is setting the join date to the current system date
and assigning appropriate values to the created_by and date_created fields.
-- If the record is being updated, appropriate values are assigned to the modified_by and
modified_date fields.

DELIMITER //

CREATE TRIGGER TRG_Application_Update
            BEFORE UPDATE ON Application
            FOR EACH ROW
    BEGIN
                        IF NEW.modified_by IS NULL THEN
                                    SET NEW.modified_by := CURRENT_USER();
                        END IF;
        IF NEW.date_modified IS NULL THEN
```

```
                        SET NEW.date_modified := NOW();
                END IF;
END;
//
DELIMITER;

-- Business purpose: The TRG_Document_Insert trigger automatically assigns a sequential
document ID to a newly-inserted row in the Document table.

DELIMITER //

CREATE TRIGGER TRG_Document_Insert
        BEFORE INSERT ON Document
        FOR EACH ROW
   BEGIN
                IF NEW.document_id IS NULL THEN
                        SET NEW.document_id := (SELECT
COALESCE(MAX(document_id), 0) + 1 FROM Document);
                END IF;
      IF NEW.created_by IS NULL THEN
                        SET NEW.created_by := CURRENT_USER();
                END IF;
      IF NEW.date_created IS NULL THEN
                        SET NEW.date_created := CURDATE();
                END IF;
END;
//
DELIMITER;
-- Business rule: TRG_Document_Update is setting the join date to the current system date
and assigning appropriate values to the created_by and date_created fields.
```

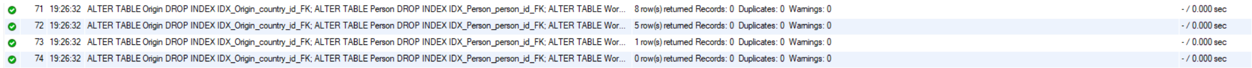
```
-- If the record is being updated, appropriate values are assigned to the modified_by and modified_date fields.

DELIMITER //

CREATE TRIGGER TRG_Document_Update
        BEFORE UPDATE ON Document
        FOR EACH ROW
   BEGIN
                IF NEW.modified_by IS NULL THEN
                        SET NEW.modified_by := CURRENT_USER();
                END IF;
      IF NEW.date_modified IS NULL THEN
                        SET NEW.date_modified := NOW();
                END IF;
END;
//

-- drop indices
ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK;
ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK;
ALTER TABLE Worker DROP INDEX IDX_Worker_worker_id_FK;
ALTER TABLE Person DROP INDEX IDX_Person_person_name;

SHOW TABLES;

SHOW FULL TABLES WHERE TABLE_TYPE = 'VIEW';
```



```
#   Time      Action                                                                                                                   Message                                                                          Duration / Fetch
1   19:26:31  DROP DATABASE IF EXISTS Immigrationdb                                                                                    10 row(s) affected                                                               0.031 sec
2   19:26:31  CREATE DATABASEIF NOT EXISTS Immigrationdb                                                                               1 row(s) affected                                                                0.000 sec
3   19:26:31  USE Immigrationdb                                                                                                        0 row(s) affected                                                                0.000 sec
4   19:26:31  DROP TABLE IF EXISTS Origin                                                                                              0 row(s) affected, 1 warning(s): 1051 Unknown table 'immigrationdb.origin'       0.000 sec
5   19:26:31  DROP TABLE IF EXISTS Person                                                                                              0 row(s) affected, 1 warning(s): 1051 Unknown table 'immigrationdb.person'       0.000 sec
6   19:26:31  DROP TABLE IF EXISTS Worker                                                                                              0 row(s) affected, 1 warning(s): 1051 Unknown table 'immigrationdb.worker'       0.000 sec
7   19:26:31  DROP TABLE IF EXISTS Application                                                                                         0 row(s) affected, 1 warning(s): 1051 Unknown table 'immigrationdb.application'  0.016 sec
8   19:26:31  DROP TABLE IF EXISTS Document                                                                                            0 row(s) affected, 1 warning(s): 1051 Unknown table 'immigrationdb.document'     0.000 sec
9   19:26:31  DROP VIEW IF EXISTS OriginInfo                                                                                           0 row(s) affected, 1 warning(s): 1051 Unknown table 'immigrationdb.origininfo'   0.000 sec
10  19:26:31  DROP VIEW IF EXISTS PersonInfo                                                                                           0 row(s) affected, 1 warning(s): 1051 Unknown table 'immigrationdb.personinfo'   0.000 sec
11  19:26:31  DROP VIEW IF EXISTS WorkerInfo                                                                                           0 row(s) affected, 1 warning(s): 1051 Unknown table 'immigrationdb.workerinfo'   0.000 sec
12  19:26:31  DROP VIEW IF EXISTS ApplicationInfo                                                                                      0 row(s) affected, 1 warning(s): 1051 Unknown table 'immigrationdb.applicationinfo'  0.000 sec
13  19:26:31  DROP VIEW IF EXISTS DocumentInfo                                                                                         0 row(s) affected, 1 warning(s): 1051 Unknown table 'immigrationdb.documentinfo' 0.000 sec
14  19:26:31  DROP TRIGGER IF EXISTS TRG_Origin_Insert                                                                                 0 row(s) affected, 1 warning(s): 1360 Trigger does not exist                     0.000 sec
15  19:26:31  DROP TRIGGER IF EXISTS TRG_Origin_Update                                                                                 0 row(s) affected, 1 warning(s): 1360 Trigger does not exist                     0.000 sec
16  19:26:31  DROP TRIGGER IF EXISTS TRG_Person_Insert                                                                                 0 row(s) affected, 1 warning(s): 1360 Trigger does not exist                     0.000 sec
17  19:26:31  DROP TRIGGER IF EXISTS TRG_Person_Update                                                                                 0 row(s) affected, 1 warning(s): 1360 Trigger does not exist                     0.000 sec
18  19:26:31  DROP TRIGGER IF EXISTS TRG_Worker_Insert                                                                                 0 row(s) affected, 1 warning(s): 1360 Trigger does not exist                     0.000 sec
19  19:26:31  DROP TRIGGER IF EXISTS TRG_Worker_Update                                                                                 0 row(s) affected, 1 warning(s): 1360 Trigger does not exist                     0.000 sec
20  19:26:31  DROP TRIGGER IF EXISTS TRG_Application_Insert                                                                            0 row(s) affected, 1 warning(s): 1360 Trigger does not exist                     0.000 sec
21  19:26:31  DROP TRIGGER IF EXISTS TRG_Application_Update                                                                            0 row(s) affected, 1 warning(s): 1360 Trigger does not exist                     0.000 sec
22  19:26:31  DROP TRIGGER IF EXISTS TRG_Document_Insert                                                                               0 row(s) affected, 1 warning(s): 1360 Trigger does not exist                     0.000 sec
23  19:26:31  DROP TRIGGER IF EXISTS TRG_Document_Update                                                                               0 row(s) affected, 1 warning(s): 1360 Trigger does not exist                     0.000 sec
24  19:26:31  CREATE TABLE Origin (   country_id        INTEGER     NOT NULL AUTO_INCREMENT,    country_name        VARCHAR(30)  NOT NULL,  ...  0 row(s) affected                      0.000 sec
25  19:26:31  CREATE TABLE Person (   person_id        INTEGER NOT NULLAUTO_INCREMENT,    country_idINTEGERNOT NULL,    person_name        ...  0 row(s) affected                        0.015 sec
26  19:26:31  CREATE TABLE Worker (   worker_id        INTEGER     NOT NULL AUTO_INCREMENT,    worker_name        VARCHAR(30)  NOT NULL,  ...  0 row(s) affected                        0.000 sec
27  19:26:31  CREATE TABLE Application (   application_id        INTEGER     NOT NULL AUTO_INCREMENT,    person_id          INTEGER     NOT NULL,...  0 row(s) affected          0.015 sec
28  19:26:31  CREATE TABLE Document (   document_id         INTEGER     NOT NULL AUTO_INCREMENT,    application_id  INTEGERNOT NULL,    doc_...  0 row(s) affected                0.000 sec
29  19:26:32  CREATE INDEX IDX_Origin_country_id_FK ON Origin (country_id)                                                             0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0                       0.016 sec
30  19:26:32  CREATE INDEX IDX_Person_person_id_FK ON Person (person_id)                                                               0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0                       0.000 sec
31  19:26:32  CREATE INDEX IDX_Worker_worker_id_FK ON Worker (worker_id)                                                               0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0                       0.015 sec
32  19:26:32  CREATE INDEX IDX_Person_person_name ON Person (person_name)                                                              0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0                       0.016 sec
33  19:26:32  CREATE OR REPLACE VIEW OriginInfo AS SELECT country_id, country_name, diplomat_name, address, ambassador_add FROM Origin   0 row(s) affected                                                           0.016 sec
34  19:26:32  CREATE OR REPLACE VIEW PersonInfo AS SELECT  person_id, person_name, dateofbirth, gender, address  FROM Person            0 row(s) affected                                                                0.000 sec
35  19:26:32  CREATE OR REPLACE VIEW WorkerInfo AS SELECT worker_id, worker_name, contact_phone, address, email FROM Worker             0 row(s) affected                                                                0.000 sec
```



```
36  19:26:32  CREATE OR REPLACE VIEW ApplicationInfo AS SELECT application_id, person_id, worker_id, app_status, submit_date FROM Application  0 row(s) affected                                 0.000 sec
37  19:26:32  CREATE OR REPLACE VIEW DocumentInfo AS SELECT country_id, country_name, diplomat_name, address, ambassador_add FROM Origin       0 row(s) affected                                 0.015 sec
38  19:26:32  CREATE TRIGGER TRG_Origin_Insert BEFORE INSERT ON Origin FOR EACH ROW     BEGIN IF NEW.country_id IS NULL THEN SET NEW.country...  0 row(s) affected                     0.000 sec
39  19:26:32  CREATE TRIGGER TRG_Origin_Update BEFORE UPDATE ON Origin FOR EACH ROW     BEGIN IF NEW.modified_by IS NULL THEN SET NEW.mo...  0 row(s) affected                        0.000 sec
40  19:26:32  CREATE TRIGGER TRG_Person_Insert BEFORE INSERT ON Person FOR EACH ROW     BEGIN IF NEW.person_id IS NULL THEN SET NEW.perso...  0 row(s) affected                       0.000 sec
41  19:26:32  CREATE TRIGGER TRG_Person_Update BEFORE UPDATE ON Person FOR EACH ROW     BEGIN IF NEW.modified_by IS NULL THEN SET NEW....  0 row(s) affected                          0.016 sec
42  19:26:32  CREATE TRIGGER TRG_Worker_Insert BEFORE INSERT ON Worker FOR EACH ROW     BEGIN IF NEW.worker_id IS NULL THEN SET NEW.wor...  0 row(s) affected                         0.000 sec
43  19:26:32  CREATE TRIGGER TRG_Worker_Update BEFORE UPDATE ON Worker FOR EACH ROW     BEGIN IF NEW.modified_by IS NULL THEN SET NEW...  0 row(s) affected                           0.000 sec
44  19:26:32  CREATE TRIGGER TRG_Application_Insert BEFORE INSERT ON Application FOR EACH ROW     BEGIN IF NEW.worker_id IS NULL THEN SET NE...  0 row(s) affected               0.015 sec
45  19:26:32  CREATE TRIGGER TRG_Application_Update BEFORE UPDATE ON Application FOR EACH ROW     BEGIN IF NEW.modified_by IS NULL THEN SE...  0 row(s) affected                 0.000 sec
46  19:26:32  CREATE TRIGGER TRG_Document_Insert BEFORE INSERT ON Document FOR EACH ROW     BEGIN IF NEW.document_id IS NULL THEN SET N...  0 row(s) affected                     0.000 sec
47  19:26:32  CREATE TRIGGER TRG_Document_Update BEFORE UPDATE ON Document FOR EACH ROW     BEGIN IF NEW.modified_by IS NULL THEN SET ...  0 row(s) affected                      0.016 sec
48  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0   0.016 sec
49  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  10 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
50  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  5 row(s) returned Records: 0  Duplicates: 0  Warnings: 0   - / 0.000 sec
51  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  15 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
52  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  13 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
53  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  10 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
54  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  15 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
55  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  13 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
56  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  10 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
57  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  10 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
58  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  3 row(s) returned Records: 0  Duplicates: 0  Warnings: 0   - / 0.000 sec
59  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  2 row(s) returned Records: 0  Duplicates: 0  Warnings: 0   - / 0.000 sec
60  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  13 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
61  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  13 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
62  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  12 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
63  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  12 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
64  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  12 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
65  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  12 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
66  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  12 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
67  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  1 row(s) returned Records: 0  Duplicates: 0  Warnings: 0   - / 0.000 sec
68  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  5 row(s) returned Records: 0  Duplicates: 0  Warnings: 0   - / 0.000 sec
69  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  1 row(s) returned Records: 0  Duplicates: 0  Warnings: 0   - / 0.000 sec
70  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  11 row(s) returned Records: 0  Duplicates: 0  Warnings: 0  - / 0.000 sec
```



```
71  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  8 row(s) returned Records: 0  Duplicates: 0  Warnings: 0   - / 0.000 sec
72  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  5 row(s) returned Records: 0  Duplicates: 0  Warnings: 0   - / 0.000 sec
73  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  1 row(s) returned Records: 0  Duplicates: 0  Warnings: 0   - / 0.000 sec
74  19:26:32  ALTER TABLE Origin DROP INDEX IDX_Origin_country_id_FK; ALTER TABLE Person DROP INDEX IDX_Person_person_id_FK; ALTER TABLE Wor...  0 row(s) returned Records: 0  Duplicates: 0  Warnings: 0   - / 0.000 sec
```

DML Source Code Embedded and Output

```
/*DML Origin Table*/

INSERT INTO Origin (country_name, diplomat_name, address, ambassador_add)

VALUES ('Vietnam', 'Nguyen Le', '2021 Lake Dr', '2022 Pond Dr');


INSERT INTO Origin (country_name, diplomat_name, address, ambassador_add)
```

```
VALUES ('Brazil', 'Isabella Santos', '789 Copacabana Avenue', '1011 Ipanema Street');

INSERT INTO Origin (country_name, diplomat_name, address, ambassador_add)
VALUES ('France', 'Sophie Martin', '789 Rue de la Paix', '1011 Avenue des Champs-Elysées');

INSERT INTO Origin (country_name, diplomat_name, address, ambassador_add)
VALUES ('Russia', 'Ivan Petrov', '789 Red Square', '1011 Kremlin Street');

INSERT INTO Origin (country_name, diplomat_name, address, ambassador_add)
VALUES ('Australia', 'Emily Walker', '789 Sydney Road', '1011 Great Barrier Reef Avenue');

INSERT INTO Origin (country_name, diplomat_name, address, ambassador_add)
VALUES ('Mexico', 'Carlos Hernandez', '789 Mexico City Blvd', '1011 Teotihuacan Avenue');

INSERT INTO Origin (country_name, diplomat_name, address, ambassador_add)
VALUES ('Zimbabwe', 'Sipho Moyo', '789 Harare Road', '1011 Victoria Falls Street');

INSERT INTO Origin (country_name, diplomat_name, address, ambassador_add)
VALUES ('China', 'Wei Liu', '123 Forbidden City Rd', '456 Great Wall Lane');

INSERT INTO Origin (country_name, diplomat_name, address, ambassador_add)
VALUES ('Sweden', 'Erik Andersson', '123 Stockholm Street', '456 Gamla Stan Avenue');

INSERT INTO Origin (country_name, diplomat_name, address, ambassador_add)
VALUES ('Norway', 'Lars Berg', '789 Oslo Lane', '1011 Fjord View Road');

/*DML Person Table*/
INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('1','An Nguyen', '1992-09-23','female', '2023 Jg Drive', '87876');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
```

```
VALUES ('3', 'Maichael Smithos', '1985-05-12', 'male', '123 Main St', '87889');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('5', 'Emil Jahnson', '1998-11-30', 'female', '456 Elm Avenue', '87889');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('2', 'Alex Rodriguez', '1977-03-17', 'male', '789 Oak Lane', '98736');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('9', 'Sophia Williams', '2001-07-08', 'female', '1011 Pine Road', '98976');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('6', 'Daniel Amato', '1990-02-14', 'male', '1213 Maple Street', '98976');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('7', 'Olivia Maruping', '1982-09-05', 'female', '1415 Birch Avenue', '92351');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('8', 'Wewean Sheng', '1984-12-28', 'male', '1617 Cedar Lane', '87876');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('4', 'Ava Evan', '1995-06-19', 'female', '1819 Oakwood Drive', '87876');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('10', 'Ethar Wison', '2003-03-02', 'male', '2021 Elm Street', '87345');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('1','Trinh Vo', '1991-10-13','female', '2023 Reading Drive', '89876');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('8','Wei Wei', '1991-10-13','male', '3241 Cross Avenue', '89876');
```

```
INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('4', 'Ivan Petrov', '1985-07-25', 'male', '567 Kremlin Street', '56789');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('4', 'Elena Ivanova', '1990-03-18', 'female', '123 Red Square', '56789');

INSERT INTO Person (country_id, person_name, dateofbirth, gender, address, zipcode)
VALUES ('4', 'Dmitri Smirnov', '1982-11-02', 'male', '789 Volga Lane', '67890');



/*DML Worker Table*/

INSERT INTO Worker (worker_name, contact_phone, address, email)
VALUES ('Charlie Coffey', '888-342-1234', '789 Osward Lane', 'c.coffey@agent_gov.us');

INSERT INTO Worker (worker_name, contact_phone, address, email)
VALUES ('Sophia Smith', '888-342-1235', '789 Osward Lane', 's.smith@agent_gov.us');

INSERT INTO Worker (worker_name, contact_phone, address, email)
VALUES ('Liam Johnson', '888-342-1236', '789 Osward Lane', 'l.johnson@agent_gov.us');

INSERT INTO Worker (worker_name, contact_phone, address, email)
VALUES ('Olivia Martinez', '888-342-1237', '789 Osward Lane', 'o.martinez@agent_gov.us');

INSERT INTO Worker (worker_name, contact_phone, address, email)
VALUES ('Noah Davis', '888-342-1238', '789 Osward Lane', 'n.davis@agent_gov.us');

INSERT INTO Worker (worker_name, contact_phone, address, email)
VALUES ('Emma Anderson', '888-342-1239', '789 Osward Lane',
'e.anderson@agent_gov.us');
```

```
INSERT INTO Worker (worker_name, contact_phone, address, email)
VALUES ('William Taylor', '888-342-1240', '789 Osward Lane', 'w.taylor@agent_gov.us');

INSERT INTO Worker (worker_name, contact_phone, address, email)
VALUES ('Ava Jackson', '888-342-1241', '789 Osward Lane', 'a.jackson@agent_gov.us');

INSERT INTO Worker (worker_name, contact_phone, address, email)
VALUES ('James Brown', '888-342-1242', '789 Osward Lane', 'j.brown@agent_gov.us');

INSERT INTO Worker (worker_name, contact_phone, address, email)
VALUES ('Isabella White', '888-342-1243', '789 Osward Lane', 'i.white@agent_gov.us');

/*DML Application Table*/

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('1','1','PENDING', '2023-01-23');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('2', '2', 'REVIEWED', '2022-03-15');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('3', '3', 'APPROVED', '2022-05-27');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('4', '4', 'PENDING', '2022-07-08');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('5', '5', 'APPROVED', '2022-09-19');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
```

```
VALUES ('6', '6', 'APPROVED', '2022-11-30');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('7', '7', 'PENDING', '2023-02-10');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('8', '8', 'REVIEWED', '2023-04-23');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('9', '9', 'APPROVED', '2023-06-05');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('10', '10', 'PENDING', '2023-07-17');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('11','1','PENDING', '2023-05-23');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('12','3','PENDING', '2023-05-23');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('13','3','REVIEWED', '2022-05-23');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('14','8','REVIEWED', '2023-06-23');

INSERT INTO Application (person_id, worker_id, app_status, submit_date)
VALUES ('15','8','REVIEWED', '2023-06-23');

/*DML Document Table*/
```

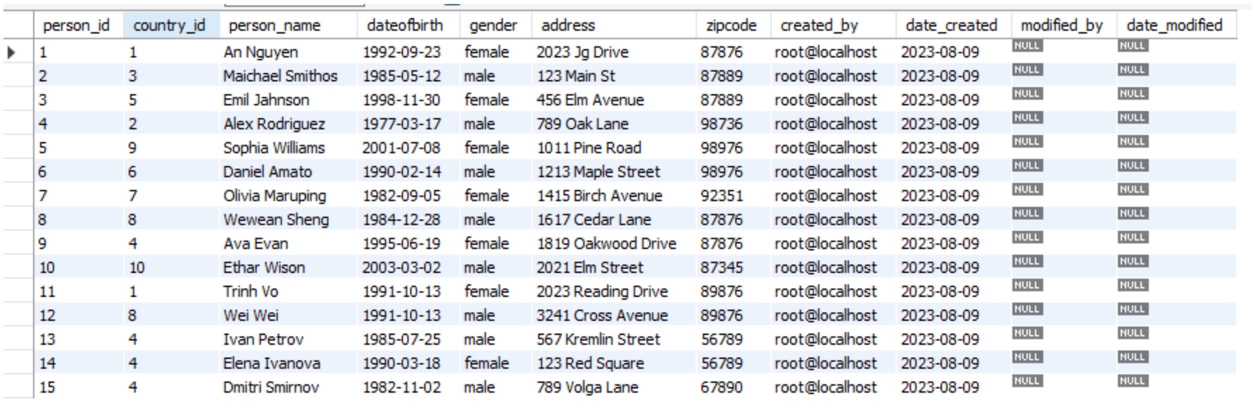
```
INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('1','Birth Certificate', 'Proof of birth', NULL , '1');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('1','Passport', 'Travel document', '2024-09-23' , '1');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('2','Birth Certificate', 'Proof of birth', NULL , '2');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('3','Diploma', 'Educational diploma', NULL, '3');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('4','Government-Issued Identification', 'Official ID', '2026-08-20', '4');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('5', 'Passport', 'Travel document', '204-07-08', '5');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('6', 'Marriage Certificate', 'Proof of marriage', NULL , '6');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('7', 'Diploma', 'Educational diploma', NULL, '7');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('8', 'Government-Issued Identification', 'Official ID', '2024-12-28', '8');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('9', 'Passport', 'Travel document', '2029-04-30', '9');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
```

```
VALUES ('11','Passport', 'Travel document', '2021-11-12' , '1');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('7','Passport', 'Travel document', '2024-11-12' , '7');

INSERT INTO Document (application_id, doc_name, doc_descr, expiration_date, country_id)
VALUES ('8','Passport', 'Travel document', '2024-11-12' , '7');

COMMIT;
```

Query Source Code Embedded and Output

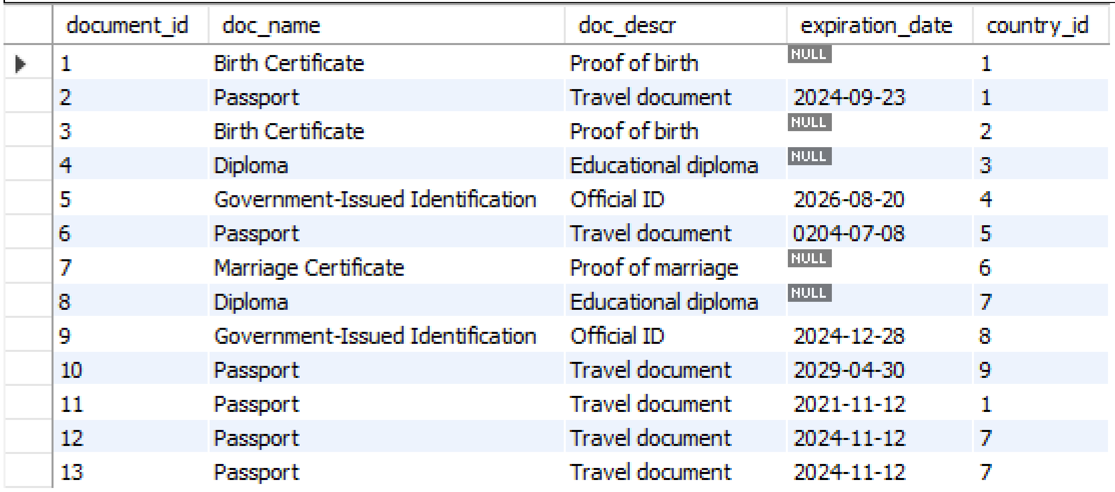
20 SQL Queries

```
-- Q1. Select all columns and all rows from one table
SELECT * FROM Person;
```



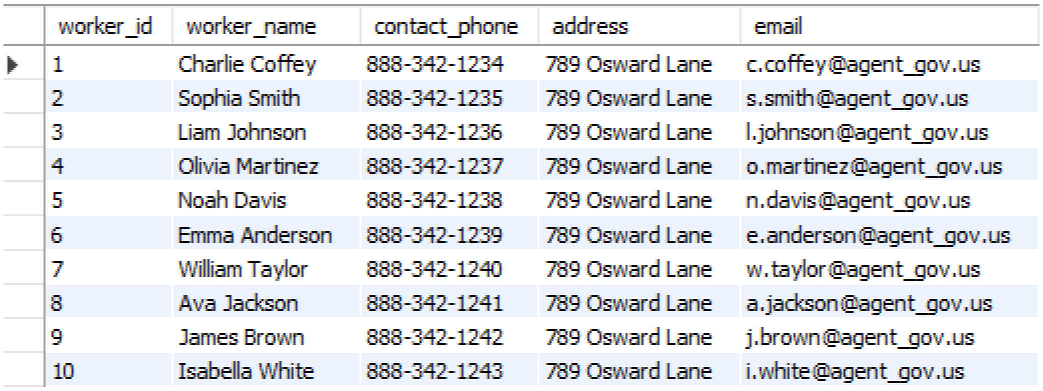
	person_id	country_id	person_name	dateofbirth	gender	address	zipcode	created_by	date_created	modified_by	date_modified
▶	1	1	An Nguyen	1992-09-23	female	2023 Jg Drive	87876	root@localhost	2023-08-09	NULL	NULL
	2	3	Maichael Smithos	1985-05-12	male	123 Main St	87889	root@localhost	2023-08-09	NULL	NULL
	3	5	Emil Jahnson	1998-11-30	female	456 Elm Avenue	87889	root@localhost	2023-08-09	NULL	NULL
	4	2	Alex Rodriguez	1977-03-17	male	789 Oak Lane	98736	root@localhost	2023-08-09	NULL	NULL
	5	9	Sophia Williams	2001-07-08	female	1011 Pine Road	98976	root@localhost	2023-08-09	NULL	NULL
	6	6	Daniel Amato	1990-02-14	male	1213 Maple Street	98976	root@localhost	2023-08-09	NULL	NULL
	7	7	Olivia Maruping	1982-09-05	female	1415 Birch Avenue	92351	root@localhost	2023-08-09	NULL	NULL
	8	8	Wewean Sheng	1984-12-28	male	1617 Cedar Lane	87876	root@localhost	2023-08-09	NULL	NULL
	9	4	Ava Evan	1995-06-19	female	1819 Oakwood Drive	87876	root@localhost	2023-08-09	NULL	NULL
	10	10	Ethar Wison	2003-03-02	male	2021 Elm Street	87345	root@localhost	2023-08-09	NULL	NULL
	11	1	Trinh Vo	1991-10-13	female	2023 Reading Drive	89876	root@localhost	2023-08-09	NULL	NULL
	12	8	Wei Wei	1991-10-13	male	3241 Cross Avenue	89876	root@localhost	2023-08-09	NULL	NULL
	13	4	Ivan Petrov	1985-07-25	male	567 Kremlin Street	56789	root@localhost	2023-08-09	NULL	NULL
	14	4	Elena Ivanova	1990-03-18	female	123 Red Square	56789	root@localhost	2023-08-09	NULL	NULL
	15	4	Dmitri Smirnov	1982-11-02	male	789 Volga Lane	67890	root@localhost	2023-08-09	NULL	NULL

```
-- Q2. Select five columns and all rows from one table
SELECT document_id, doc_name, doc_descr, expiration_date, country_id
FROM Document;
```



	document_id	doc_name	doc_descr	expiration_date	country_id
▶	1	Birth Certificate	Proof of birth	NULL	1
	2	Passport	Travel document	2024-09-23	1
	3	Birth Certificate	Proof of birth	NULL	2
	4	Diploma	Educational diploma	NULL	3
	5	Government-Issued Identification	Official ID	2026-08-20	4
	6	Passport	Travel document	0204-07-08	5
	7	Marriage Certificate	Proof of marriage	NULL	6
	8	Diploma	Educational diploma	NULL	7
	9	Government-Issued Identification	Official ID	2024-12-28	8
	10	Passport	Travel document	2029-04-30	9
	11	Passport	Travel document	2021-11-12	1
	12	Passport	Travel document	2024-11-12	7
	13	Passport	Travel document	2024-11-12	7

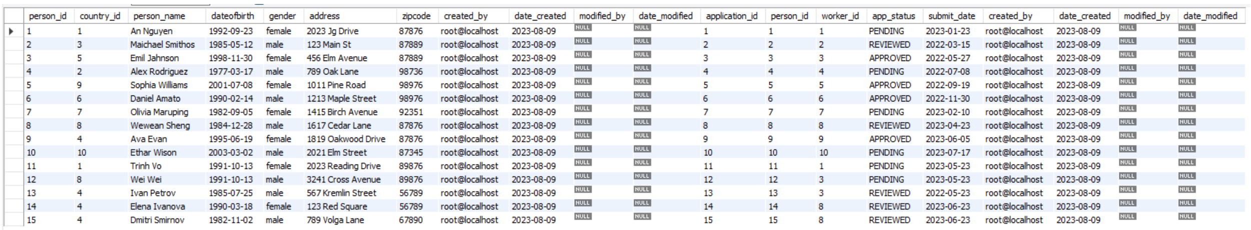
```
-- Q3. Select all columns from all rows from one view
SELECT * FROM workerinfo;
```



worker_id	worker_name	contact_phone	address	email
1	Charlie Coffey	888-342-1234	789 Osward Lane	c.coffey@agent_gov.us
2	Sophia Smith	888-342-1235	789 Osward Lane	s.smith@agent_gov.us
3	Liam Johnson	888-342-1236	789 Osward Lane	l.johnson@agent_gov.us
4	Olivia Martinez	888-342-1237	789 Osward Lane	o.martinez@agent_gov.us
5	Noah Davis	888-342-1238	789 Osward Lane	n.davis@agent_gov.us
6	Emma Anderson	888-342-1239	789 Osward Lane	e.anderson@agent_gov.us
7	William Taylor	888-342-1240	789 Osward Lane	w.taylor@agent_gov.us
8	Ava Jackson	888-342-1241	789 Osward Lane	a.jackson@agent_gov.us
9	James Brown	888-342-1242	789 Osward Lane	j.brown@agent_gov.us
10	Isabella White	888-342-1243	789 Osward Lane	i.white@agent_gov.us

-- Q4. Using a join on 2 tables, select all columns and all rows from the tables without the use of a Cartesian product.

SELECT * FROM Person LEFT OUTER JOIN Application ON Person.person_id = Application.person_id;

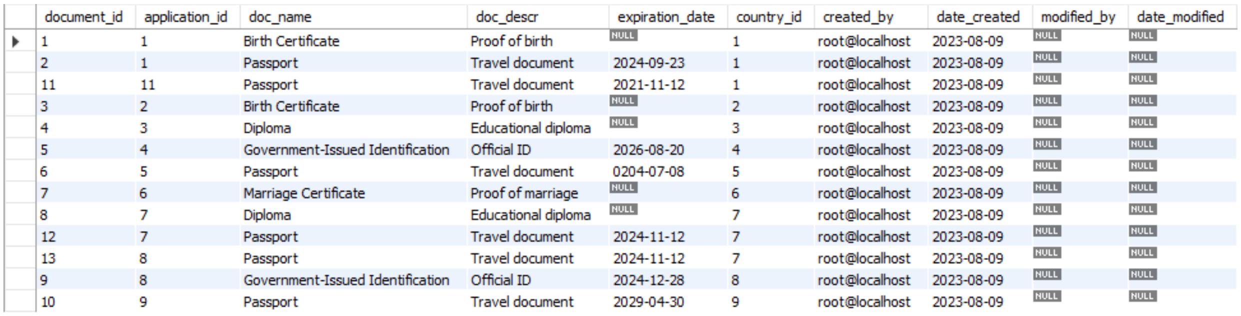


person_id	country_id	person_name	dateofbirth	gender	address	zipcode	created_by	date_created	modified_by	date_modified	application_id	person_id	worker_id	app_status	submit_date	created_by	date_created	modified_by	date_modified
1	1	An Nguyen	1992-09-23	female	2023 Jg Drive	87876	root@localhost	2023-08-09	NULL	NULL	1	1	1	PENDING	2023-01-23	root@localhost	2023-08-09	NULL	NULL
2	3	Maichael Smithos	1985-05-12	male	123 Main St	87889	root@localhost	2023-08-09	NULL	NULL	2	2	2	REVIEWED	2022-03-15	root@localhost	2023-08-09	NULL	NULL
3	5	Emil Jahnson	1998-11-30	female	456 Elm Avenue	87889	root@localhost	2023-08-09	NULL	NULL	3	3	3	APPROVED	2022-05-27	root@localhost	2023-08-09	NULL	NULL
4	2	Alex Rodriguez	1977-03-17	male	789 Oak Lane	98736	root@localhost	2023-08-09	NULL	NULL	4	4	4	PENDING	2022-07-08	root@localhost	2023-08-09	NULL	NULL
5	9	Sophia Williams	2001-07-08	female	1011 Pine Road	98976	root@localhost	2023-08-09	NULL	NULL	5	5	5	APPROVED	2022-09-19	root@localhost	2023-08-09	NULL	NULL
6	6	Daniel Amato	1990-02-14	male	1213 Maple Street	98976	root@localhost	2023-08-09	NULL	NULL	6	6	6	APPROVED	2022-11-30	root@localhost	2023-08-09	NULL	NULL
7	7	Olivia Maruping	1982-09-05	female	1415 Birch Avenue	92351	root@localhost	2023-08-09	NULL	NULL	7	7	7	PENDING	2023-02-10	root@localhost	2023-08-09	NULL	NULL
8	8	Wewean Sheng	1984-12-28	male	1617 Cedar Lane	87876	root@localhost	2023-08-09	NULL	NULL	8	8	8	REVIEWED	2023-04-23	root@localhost	2023-08-09	NULL	NULL
9	4	Ava Evan	1995-06-19	female	1819 Oakwood Drive	87876	root@localhost	2023-08-09	NULL	NULL	9	9	9	APPROVED	2023-06-05	root@localhost	2023-08-09	NULL	NULL
10	10	Ethar Wison	2003-03-02	male	2021 Elm Street	87345	root@localhost	2023-08-09	NULL	NULL	10	10	10	PENDING	2023-07-17	root@localhost	2023-08-09	NULL	NULL
11	1	Trinh Vo	1991-10-13	female	2023 Reading Drive	89876	root@localhost	2023-08-09	NULL	NULL	11	11	1	PENDING	2023-05-23	root@localhost	2023-08-09	NULL	NULL
12	8	Wei Wei	1991-10-13	male	3241 Cross Avenue	89876	root@localhost	2023-08-09	NULL	NULL	12	12	3	PENDING	2023-05-23	root@localhost	2023-08-09	NULL	NULL
13	4	Ivan Petrov	1985-07-25	male	567 Kremlin Street	56789	root@localhost	2023-08-09	NULL	NULL	13	13	3	REVIEWED	2022-05-23	root@localhost	2023-08-09	NULL	NULL
14	4	Elena Ivanova	1990-03-18	female	123 Red Square	56789	root@localhost	2023-08-09	NULL	NULL	14	14	8	REVIEWED	2023-06-23	root@localhost	2023-08-09	NULL	NULL
15	4	Dmitri Smirnov	1982-11-02	male	789 Volga Lane	67890	root@localhost	2023-08-09	NULL	NULL	15	15	8	REVIEWED	2023-06-23	root@localhost	2023-08-09	NULL	NULL

-- Q5. Select and order data retrieved from one table

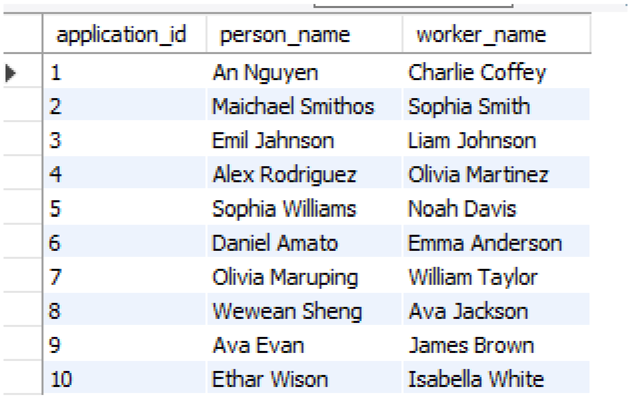
SELECT * FROM Document

ORDER BY country_id;



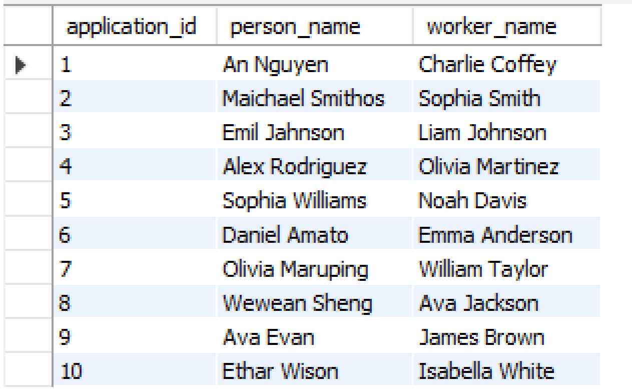
document_id	application_id	doc_name	doc_descr	expiration_date	country_id	created_by	date_created	modified_by	date_modified
1	1	Birth Certificate	Proof of birth	NULL	1	root@localhost	2023-08-09	NULL	NULL
2	1	Passport	Travel document	2024-09-23	1	root@localhost	2023-08-09	NULL	NULL
11	11	Passport	Travel document	2021-11-12	1	root@localhost	2023-08-09	NULL	NULL
3	2	Birth Certificate	Proof of birth	NULL	2	root@localhost	2023-08-09	NULL	NULL
4	3	Diploma	Educational diploma	NULL	3	root@localhost	2023-08-09	NULL	NULL
5	4	Government-Issued Identification	Official ID	2026-08-20	4	root@localhost	2023-08-09	NULL	NULL
6	5	Passport	Travel document	0204-07-08	5	root@localhost	2023-08-09	NULL	NULL
7	6	Marriage Certificate	Proof of marriage	NULL	6	root@localhost	2023-08-09	NULL	NULL
8	7	Diploma	Educational diploma	NULL	7	root@localhost	2023-08-09	NULL	NULL
12	7	Passport	Travel document	2024-11-12	7	root@localhost	2023-08-09	NULL	NULL
13	8	Passport	Travel document	2024-11-12	7	root@localhost	2023-08-09	NULL	NULL
9	8	Government-Issued Identification	Official ID	2024-12-28	8	root@localhost	2023-08-09	NULL	NULL
10	9	Passport	Travel document	2029-04-30	9	root@localhost	2023-08-09	NULL	NULL

```
-- Q6. Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 10 rows
SELECT A.application_id, P.person_name, W.worker_name
FROM Application A
JOIN Person P ON A.person_id = P.person_id
JOIN Worker W ON A.worker_id = W.worker_id
ORDER BY A.application_id
LIMIT 10;
```



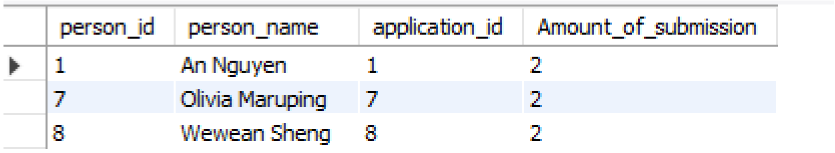
application_id	person_name	worker_name
1	An Nguyen	Charlie Coffey
2	Maichael Smithos	Sophia Smith
3	Emil Jahnson	Liam Johnson
4	Alex Rodriguez	Olivia Martinez
5	Sophia Williams	Noah Davis
6	Daniel Amato	Emma Anderson
7	Olivia Maruping	William Taylor
8	Wewean Sheng	Ava Jackson
9	Ava Evan	James Brown
10	Ethar Wison	Isabella White

```
-- Q7. Select distinct rows using joins on 3 tables
SELECT DISTINCT A.application_id, P.person_name, W.worker_name
FROM Application A
JOIN Person P ON A.person_id = P.person_id
JOIN Worker W ON A.worker_id = W.worker_id
ORDER BY A.application_id
LIMIT 10;
```



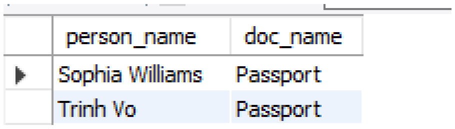
application_id	person_name	worker_name
1	An Nguyen	Charlie Coffey
2	Maichael Smithos	Sophia Smith
3	Emil Jahnson	Liam Johnson
4	Alex Rodriguez	Olivia Martinez
5	Sophia Williams	Noah Davis
6	Daniel Amato	Emma Anderson
7	Olivia Maruping	William Taylor
8	Wewean Sheng	Ava Jackson
9	Ava Evan	James Brown
10	Ethar Wison	Isabella White

```
-- Q8. Use GROUP BY and HAVING in a select statement using one or more tables
-- Display the name of person, person_id and application_id who have submit at least 2 documents
SELECT P.person_id, P.person_name, A.application_id, COUNT(D.document_id) AS Amount_of_submission
FROM Person P
JOIN Application A ON P.person_id = A.person_id
JOIN Document D ON A.application_id = D.application_id
GROUP BY P.person_id, P.person_name, A.application_id
HAVING COUNT(D.document_id) >= 2;
```



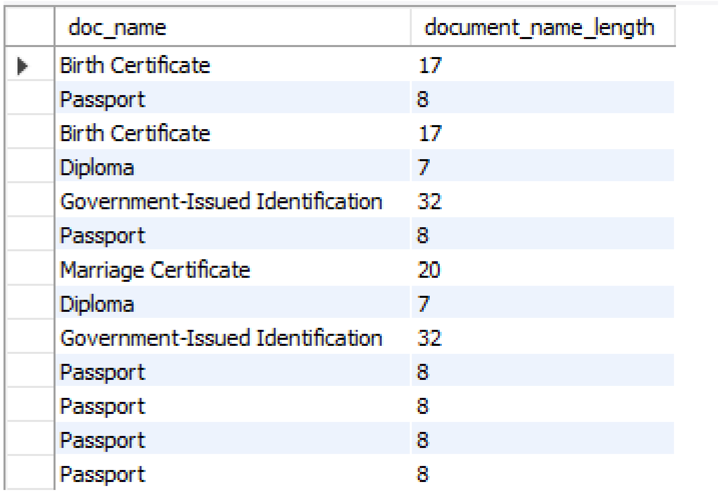
person_id	person_name	application_id	Amount_of_submission
1	An Nguyen	1	2
7	Olivia Maruping	7	2
8	Wewean Sheng	8	2

```
-- Q9. Use IN clause to select data from one or more tables
-- Display the name of person, person_id and application_id who have submit expired
document
SELECT P.person_name, D.doc_name
FROM Person P
JOIN Application A ON P.person_id = A.person_id
JOIN Document D ON A.application_id = D.application_id
WHERE P.person_id IN (
    SELECT A.person_id
    FROM Application A
    JOIN Document D ON A.application_id = D.application_id
    WHERE D.expiration_date IS NOT NULL AND D.expiration_date < CURDATE()
);
```



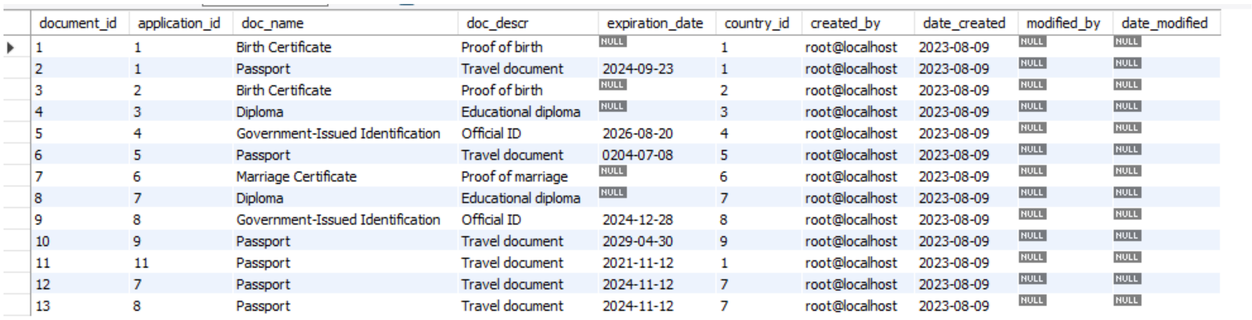
person_name	doc_name
Sophia Williams	Passport
Trinh Vo	Passport

```
-- Q10. Select length of one column from one table (use LENGTH function)
-- Display doc_name length
SELECT doc_name, LENGTH(doc_name) AS document_name_length
FROM Document;
```

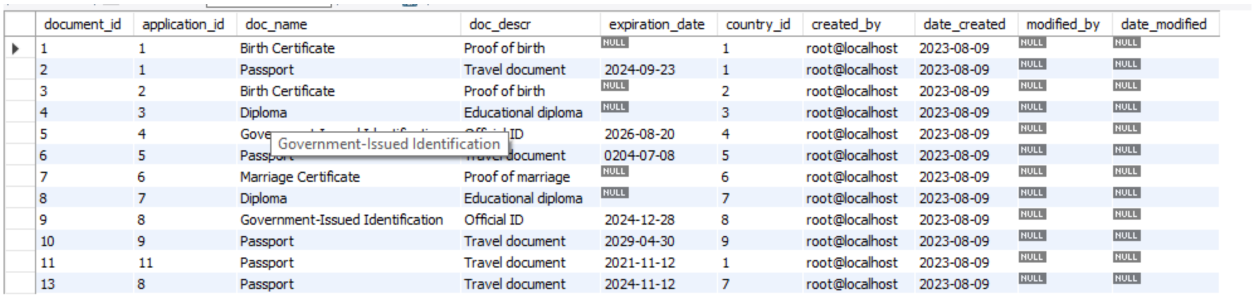


doc_name	document_name_length
Birth Certificate	17
Passport	8
Birth Certificate	17
Diploma	7
Government-Issued Identification	32
Passport	8
Marriage Certificate	20
Diploma	7
Government-Issued Identification	32
Passport	8
Passport	8
Passport	8
Passport	8

```
-- Q.11 Delete one record from one table.
-- Use select statements to demonstrate the table contents before and after the DELETE statement.
-- Make sure you use ROLLBACK afterwards so that the data will not be physically removed
-- Check table before deletion
SELECT * FROM Document;
-- Delete the record
DELETE FROM Document WHERE document_id = 12;
-- Show table after deletion has been comitted
SELECT * FROM Document;
-- Rollback the transaction to restore the deleted record
ROLLBACK;
-- Check table for Rollback
SELECT * FROM Document;
```

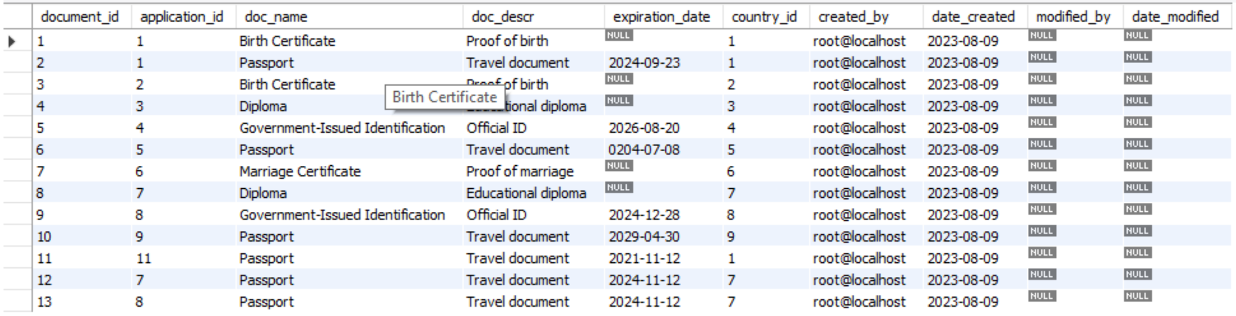


document_id	application_id	doc_name	doc_descr	expiration_date	country_id	created_by	date_created	modified_by	date_modified
1	1	Birth Certificate	Proof of birth	NULL	1	root@localhost	2023-08-09	NULL	NULL
2	1	Passport	Travel document	2024-09-23	1	root@localhost	2023-08-09	NULL	NULL
3	2	Birth Certificate	Proof of birth	NULL	2	root@localhost	2023-08-09	NULL	NULL
4	3	Diploma	Educational diploma	NULL	3	root@localhost	2023-08-09	NULL	NULL
5	4	Government-Issued Identification	Official ID	2026-08-20	4	root@localhost	2023-08-09	NULL	NULL
6	5	Passport	Travel document	0204-07-08	5	root@localhost	2023-08-09	NULL	NULL
7	6	Marriage Certificate	Proof of marriage	NULL	6	root@localhost	2023-08-09	NULL	NULL
8	7	Diploma	Educational diploma	NULL	7	root@localhost	2023-08-09	NULL	NULL
9	8	Government-Issued Identification	Official ID	2024-12-28	8	root@localhost	2023-08-09	NULL	NULL
10	9	Passport	Travel document	2029-04-30	9	root@localhost	2023-08-09	NULL	NULL
11	11	Passport	Travel document	2021-11-12	1	root@localhost	2023-08-09	NULL	NULL
12	7	Passport	Travel document	2024-11-12	7	root@localhost	2023-08-09	NULL	NULL
13	8	Passport	Travel document	2024-11-12	7	root@localhost	2023-08-09	NULL	NULL

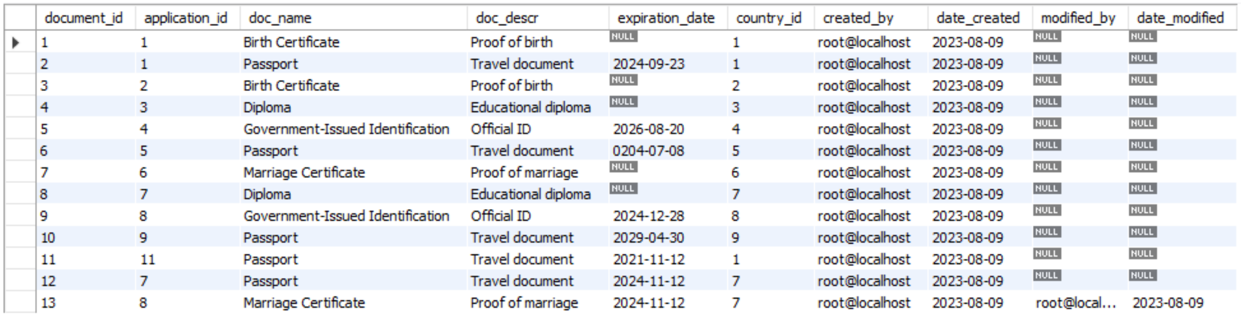


document_id	application_id	doc_name	doc_descr	expiration_date	country_id	created_by	date_created	modified_by	date_modified
1	1	Birth Certificate	Proof of birth	NULL	1	root@localhost	2023-08-09	NULL	NULL
2	1	Passport	Travel document	2024-09-23	1	root@localhost	2023-08-09	NULL	NULL
3	2	Birth Certificate	Proof of birth	NULL	2	root@localhost	2023-08-09	NULL	NULL
4	3	Diploma	Educational diploma	NULL	3	root@localhost	2023-08-09	NULL	NULL
5	4	Government-Issued Identification	Official ID	2026-08-20	4	root@localhost	2023-08-09	NULL	NULL
6	5	Passport	Travel document	0204-07-08	5	root@localhost	2023-08-09	NULL	NULL
7	6	Marriage Certificate	Proof of marriage	NULL	6	root@localhost	2023-08-09	NULL	NULL
8	7	Diploma	Educational diploma	NULL	7	root@localhost	2023-08-09	NULL	NULL
9	8	Government-Issued Identification	Official ID	2024-12-28	8	root@localhost	2023-08-09	NULL	NULL
10	9	Passport	Travel document	2029-04-30	9	root@localhost	2023-08-09	NULL	NULL
11	11	Passport	Travel document	2021-11-12	1	root@localhost	2023-08-09	NULL	NULL
13	8	Passport	Travel document	2024-11-12	7	root@localhost	2023-08-09	NULL	NULL

```
-- Q12. Update one record from one table.
-- Use select statements to demonstrate the table contents before and after the UPDATE
statement.
-- Make sure you use ROLLBACK afterwards so that the data will not be physically removed
-- Check table before update
SELECT * FROM Document;
-- Update the record
UPDATE Document
SET doc_name = 'Marriage Certificate', doc_descr = 'Proof of marriage'
WHERE document_id = 13;
-- Check table after update
SELECT * FROM Document;
-- Rollback the transaction to restore original data
ROLLBACK;
-- Check Rollback
SELECT * FROM Document;
```

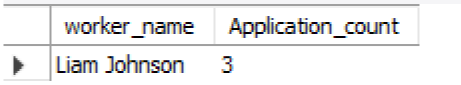


document_id	application_id	doc_name	doc_descr	expiration_date	country_id	created_by	date_created	modified_by	date_modified
1	1	Birth Certificate	Proof of birth	NULL	1	root@localhost	2023-08-09	NULL	NULL
2	1	Passport	Travel document	2024-09-23	1	root@localhost	2023-08-09	NULL	NULL
3	2	Birth Certificate	Proof of birth	NULL	2	root@localhost	2023-08-09	NULL	NULL
4	3	Diploma	Educational diploma	NULL	3	root@localhost	2023-08-09	NULL	NULL
5	4	Government-Issued Identification	Official ID	2026-08-20	4	root@localhost	2023-08-09	NULL	NULL
6	5	Passport	Travel document	0204-07-08	5	root@localhost	2023-08-09	NULL	NULL
7	6	Marriage Certificate	Proof of marriage	NULL	6	root@localhost	2023-08-09	NULL	NULL
8	7	Diploma	Educational diploma	NULL	7	root@localhost	2023-08-09	NULL	NULL
9	8	Government-Issued Identification	Official ID	2024-12-28	8	root@localhost	2023-08-09	NULL	NULL
10	9	Passport	Travel document	2029-04-30	9	root@localhost	2023-08-09	NULL	NULL
11	11	Passport	Travel document	2021-11-12	1	root@localhost	2023-08-09	NULL	NULL
12	7	Passport	Travel document	2024-11-12	7	root@localhost	2023-08-09	NULL	NULL
13	8	Passport	Travel document	2024-11-12	7	root@localhost	2023-08-09	NULL	NULL



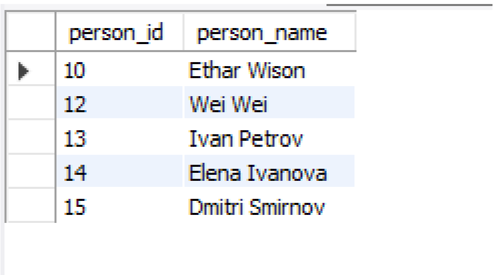
document_id	application_id	doc_name	doc_descr	expiration_date	country_id	created_by	date_created	modified_by	date_modified
1	1	Birth Certificate	Proof of birth	NULL	1	root@localhost	2023-08-09	NULL	NULL
2	1	Passport	Travel document	2024-09-23	1	root@localhost	2023-08-09	NULL	NULL
3	2	Birth Certificate	Proof of birth	NULL	2	root@localhost	2023-08-09	NULL	NULL
4	3	Diploma	Educational diploma	NULL	3	root@localhost	2023-08-09	NULL	NULL
5	4	Government-Issued Identification	Official ID	2026-08-20	4	root@localhost	2023-08-09	NULL	NULL
6	5	Passport	Travel document	0204-07-08	5	root@localhost	2023-08-09	NULL	NULL
7	6	Marriage Certificate	Proof of marriage	NULL	6	root@localhost	2023-08-09	NULL	NULL
8	7	Diploma	Educational diploma	NULL	7	root@localhost	2023-08-09	NULL	NULL
9	8	Government-Issued Identification	Official ID	2024-12-28	8	root@localhost	2023-08-09	NULL	NULL
10	9	Passport	Travel document	2029-04-30	9	root@localhost	2023-08-09	NULL	NULL
11	11	Passport	Travel document	2021-11-12	1	root@localhost	2023-08-09	NULL	NULL
12	7	Passport	Travel document	2024-11-12	7	root@localhost	2023-08-09	NULL	NULL
13	8	Marriage Certificate	Proof of marriage	2024-11-12	7	root@localhost	2023-08-09	root@local...	2023-08-09

```
-- Q13. Display the name of worker of the most application
SELECT W.worker_name, COUNT(A.application_id) AS Application_count
FROM Worker W
JOIN Application A ON W.worker_id = A.worker_id
GROUP BY W.worker_name
ORDER BY application_count DESC
LIMIT 1;
```



	worker_name	Application_count
▶	Liam Johnson	3

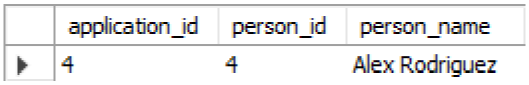
```
-- Q14. Display the name of person who have not submit any document
SELECT P.person_id, P.person_name
FROM Person P
LEFT JOIN Application A ON P.person_id = A.person_id
LEFT JOIN Document D ON A.application_id = D.application_id
WHERE D.document_id IS NULL;
```



	person_id	person_name
▶	10	Ethar Wison
	12	Wei Wei
	13	Ivan Petrov
	14	Elena Ivanova
	15	Dmitri Smirnov

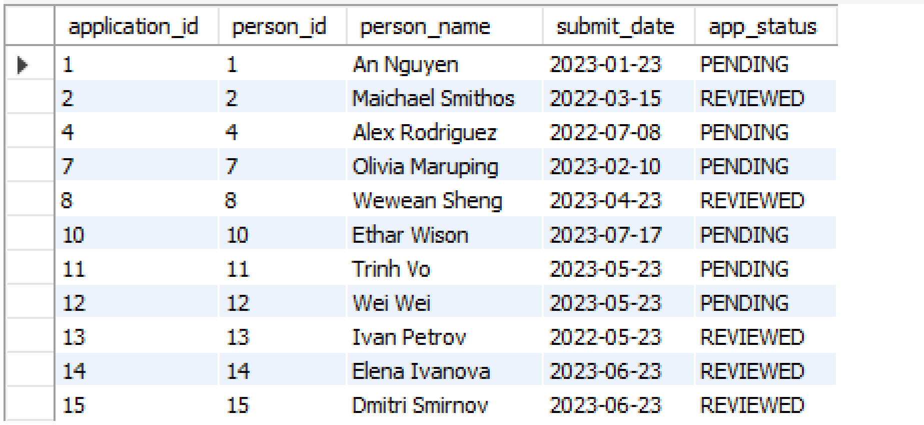
```
-- Q15. Display the name of person, person_id and application_id who have submit their
application in 2022 with Pending status
SELECT A.application_id, P.person_id, P.person_name
FROM Person P
```

```
JOIN Application A ON P.person_id = A.person_id
WHERE YEAR(A.submit_date) = 2022 AND A.app_status = 'PENDING';
```



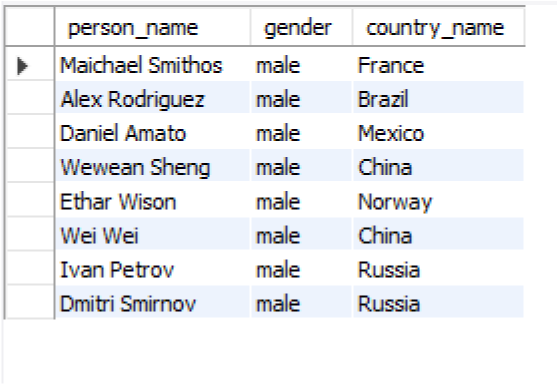
application_id	person_id	person_name
4	4	Alex Rodriguez

```
-- Q16. Display name of person, person_id, and application_id and submit date with other
status, not approved
SELECT  A.application_id, P.person_id, P.person_name, A.submit_date, A.app_status
FROM Person P
JOIN Application A ON P.person_id = A.person_id
WHERE A.app_status <> 'APPROVED';
```



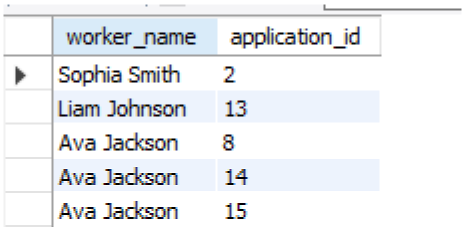
application_id	person_id	person_name	submit_date	app_status
1	1	An Nguyen	2023-01-23	PENDING
2	2	Maichael Smithos	2022-03-15	REVIEWED
4	4	Alex Rodriguez	2022-07-08	PENDING
7	7	Olivia Maruping	2023-02-10	PENDING
8	8	Wewean Sheng	2023-04-23	REVIEWED
10	10	Ethar Wison	2023-07-17	PENDING
11	11	Trinh Vo	2023-05-23	PENDING
12	12	Wei Wei	2023-05-23	PENDING
13	13	Ivan Petrov	2022-05-23	REVIEWED
14	14	Elena Ivanova	2023-06-23	REVIEWED
15	15	Dmitri Smirnov	2023-06-23	REVIEWED

```
-- Q17. Display all male applicants
SELECT P.person_name, P.gender, O.country_name
FROM Person P
JOIN Origin O ON P.country_id = O.country_id
WHERE P.gender = 'male';
```



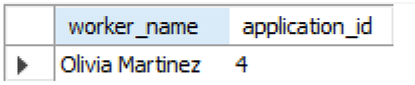
person_name	gender	country_name
Maichael Smithos	male	France
Alex Rodriguez	male	Brazil
Daniel Amato	male	Mexico
Wewean Sheng	male	China
Ethar Wison	male	Norway
Wei Wei	male	China
Ivan Petrov	male	Russia
Dmitri Smirnov	male	Russia

```
-- Q18. Display name of workers with application in reviewed
SELECT DISTINCT W.worker_name, A.application_id
FROM Worker W
JOIN Application A ON W.worker_id = A.worker_id
WHERE A.app_status = 'REVIEWED'
ORDER BY W.worker_id;
```



worker_name	application_id
Sophia Smith	2
Liam Johnson	13
Ava Jackson	8
Ava Jackson	14
Ava Jackson	15

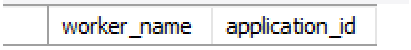
```
-- Q19. Display name of workers with application in pending status and submission date in
2022 or earlier
SELECT DISTINCT W.worker_name, A.application_id
FROM Worker W
JOIN Application A ON W.worker_id = A.worker_id
WHERE A.app_status = 'PENDING' AND YEAR(A.submit_date) <= 2022;
```



worker_name	application_id
Olivia Martinez	4

```
-- Q20. Display name of workers with application in pending status and submission date in
2022 or earlier with missing document
SELECT DISTINCT W.worker_name, A.application_id
FROM Worker W
JOIN Application A ON W.worker_id = A.worker_id
```

```
LEFT JOIN Document D ON A.application_id = D.application_id
WHERE A.app_status = 'PENDING'
  AND YEAR(A.submit_date) <= 2022
  AND D.document_id IS NULL;
```



	worker_name	application_id