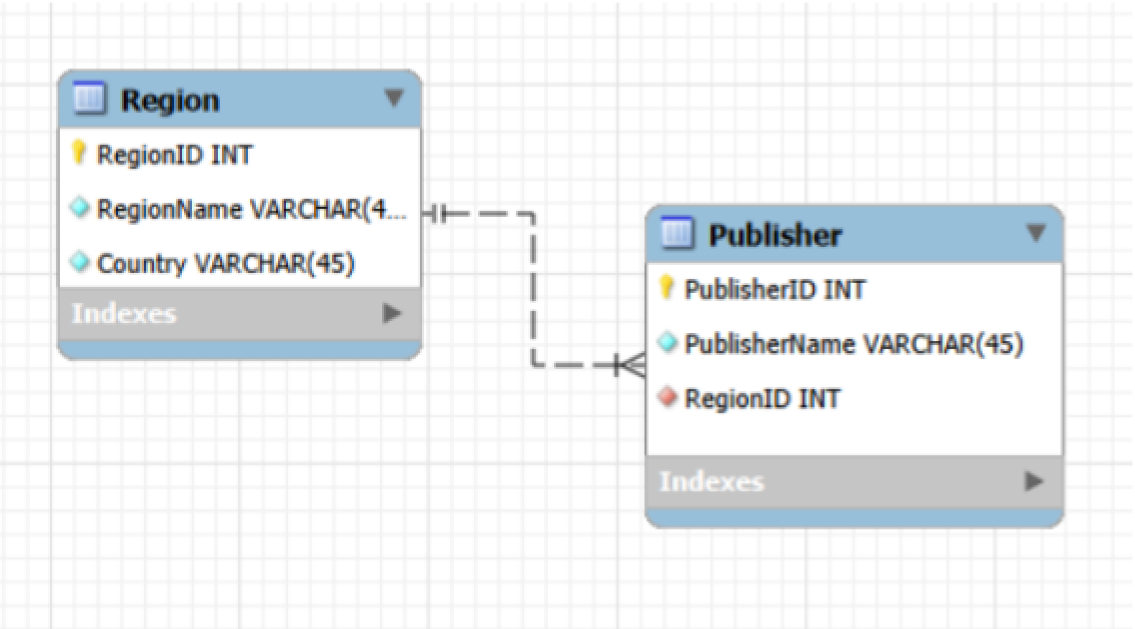
FINTECH LAB2 REPORT

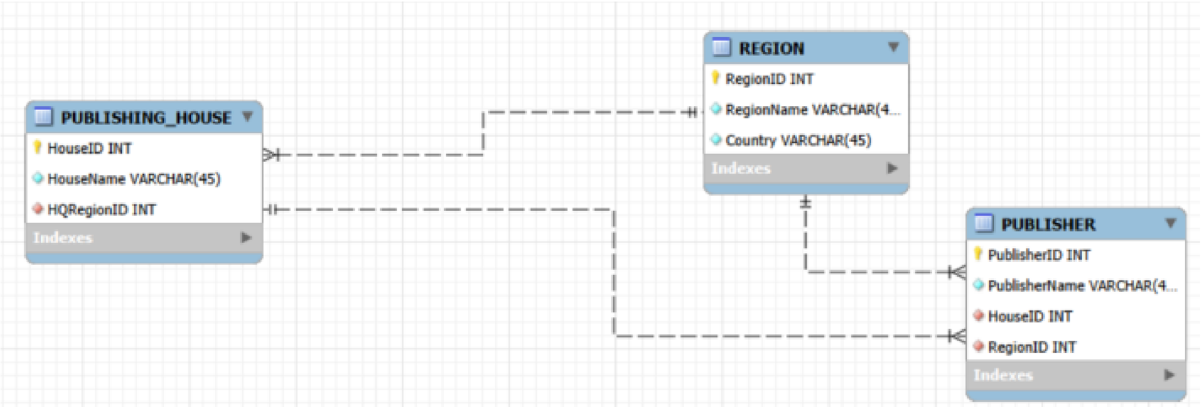
Name: Krishang Jain
Registration Number: 240958280
Roll no.: 73
Section: CSFT
Date: 17/01/2026

PPT QUESTIONS

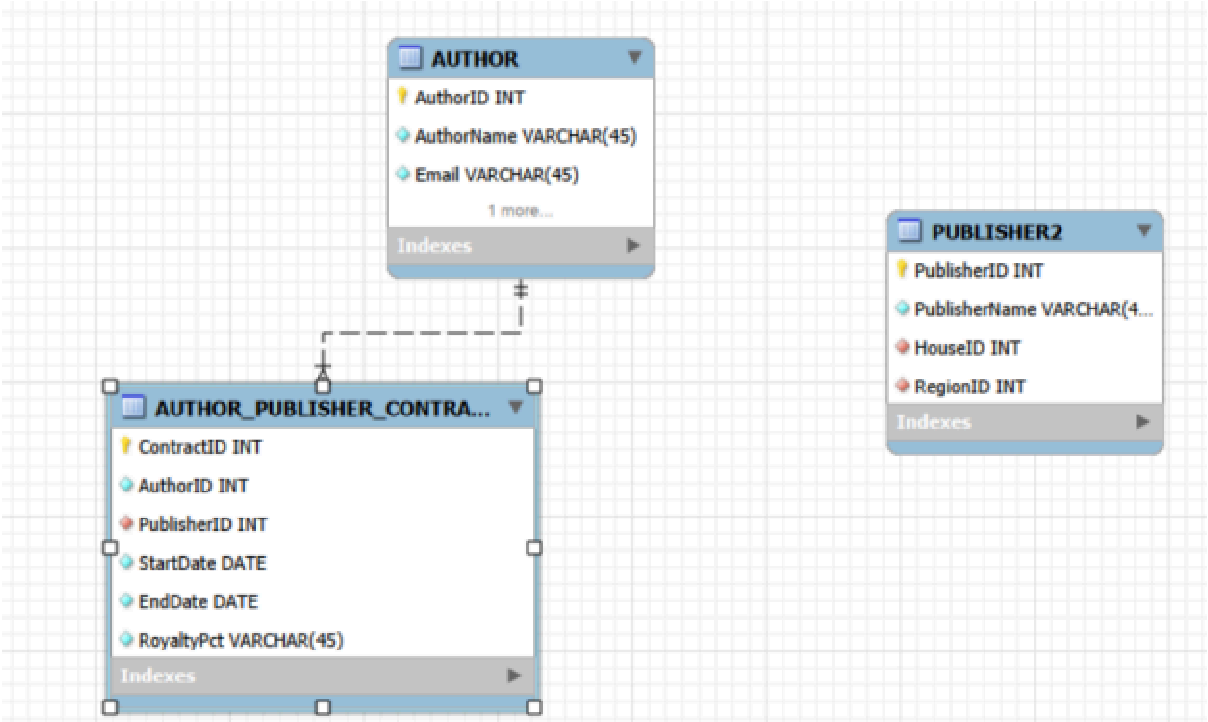
Q1:



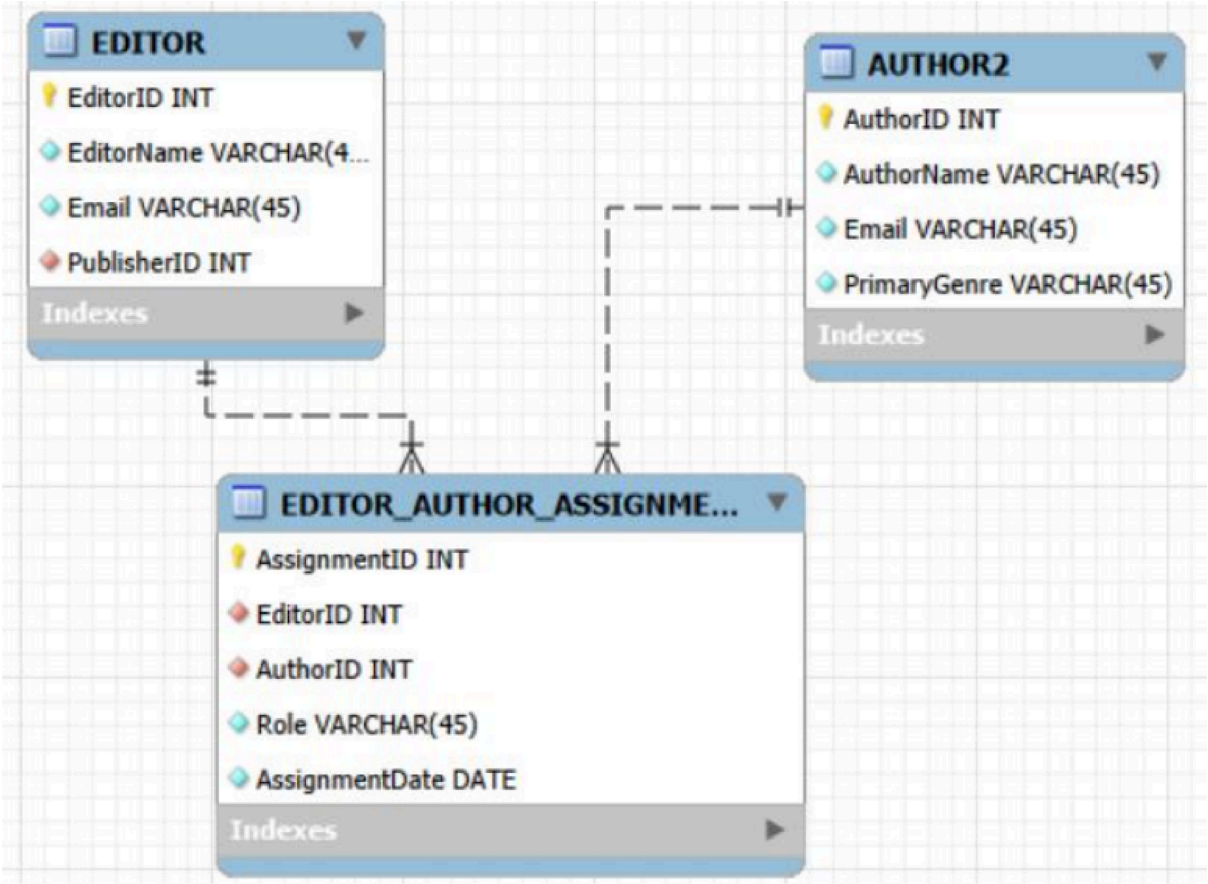
Q2:



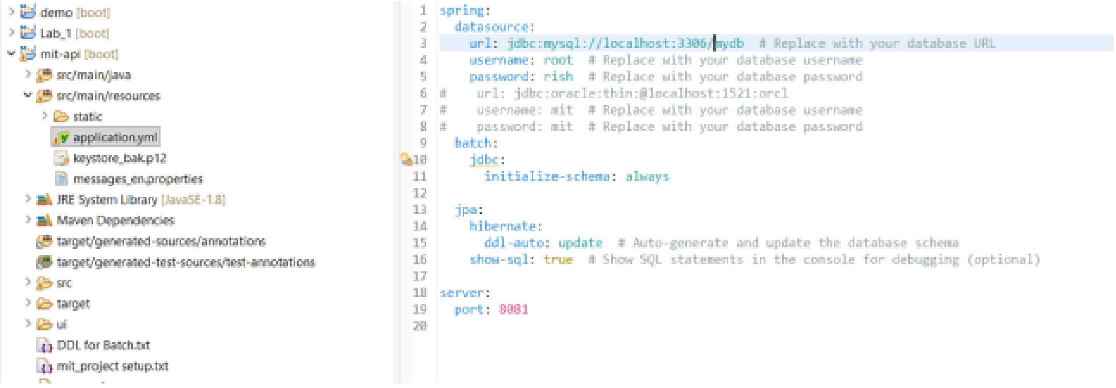
Q3: The publisher 2 table is referencing the tables in the previous ER diagrams



Q4:



SETTING UP MIT-API FOR REVERSE ENGINEERING

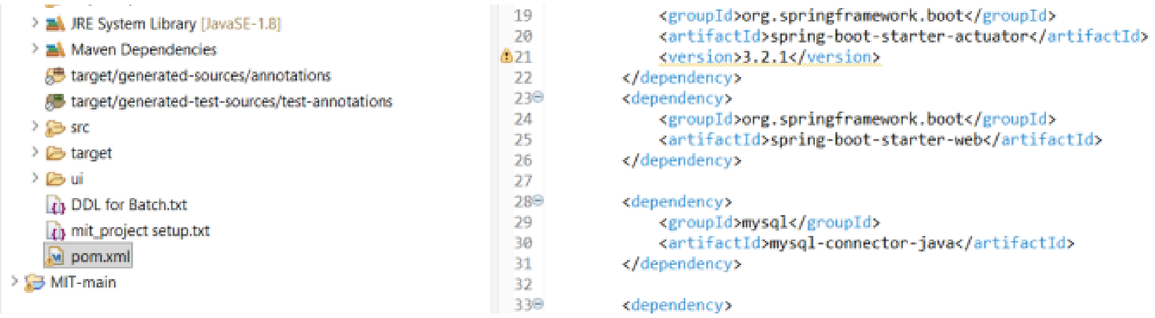


```
> demo [boot]
> Lab_1 [boot]
v mit-api [boot]
  > src/main/java
  v src/main/resources
    > static
      application.yml
      keystore_bak.p12
      messages_en.properties
  > JRE System Library [JavaSE-1.8]
  > Maven Dependencies
    target/generated-sources/annotations
    target/generated-test-sources/test-annotations
  > src
  > target
  > ui
    DDL for Batch.txt
    mit_project setup.txt
```

```yaml
spring:
  datasource:
    url: jdbc:mysql://localhost:3306/mydb  # Replace with your database URL
    username: root  # Replace with your database username
    password: rish  # Replace with your database password
#    url: jdbc:oracle:thin:@localhost:1521:orcl
#    username: mit  # Replace with your database username
#    password: mit  # Replace with your database password
  batch:
    jdbc:
      initialize-schema: always

  jpa:
    hibernate:
      ddl-auto: update  # Auto-generate and update the database schema
    show-sql: true  # Show SQL statements in the console for debugging (optional)

server:
  port: 8081
```



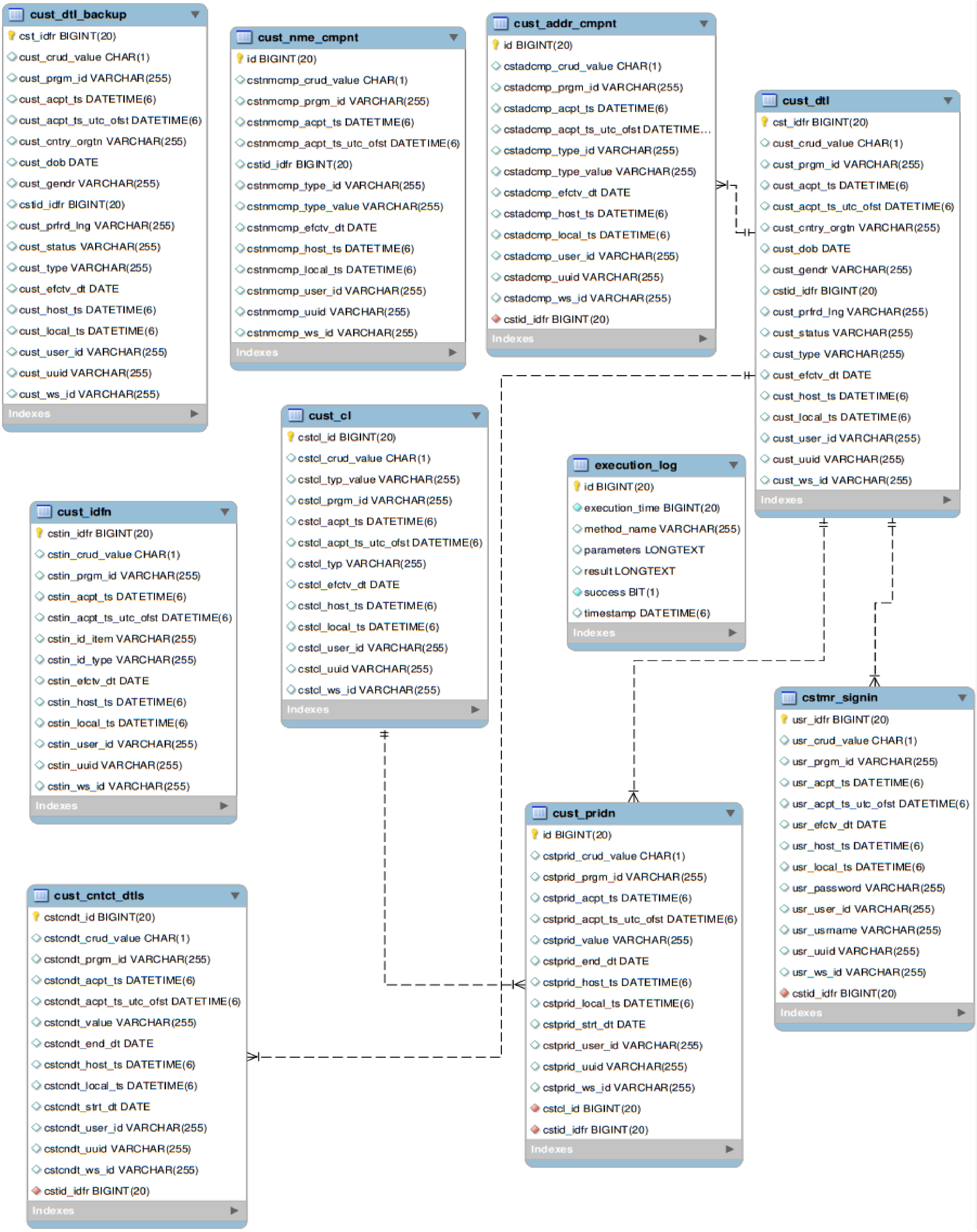
```
  > JRE System Library [JavaSE-1.8]
  > Maven Dependencies
    target/generated-sources/annotations
    target/generated-test-sources/test-annotations
  > src
  > target
  > ui
    DDL for Batch.txt
    mit_project setup.txt
    pom.xml
> MIT-main
```

```xml
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-actuator</artifactId>
            <version>3.2.1</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
        </dependency>

        <dependency>
```

Q1: Design an ERD/LDM: REVERSE ENGINEERING THE DATABASE



cust_dtl_backup

cst_idfr BIGINT(20)
cust_crud_value CHAR(1)
cust_prgm_id VARCHAR(255)
cust_acpt_ts DATETIME(6)
cust_acpt_ts_utc_ofst DATETIME(6)
cust_cntry_orgtn VARCHAR(255)
cust_dob DATE
cust_gendr VARCHAR(255)
cstid_idfr BIGINT(20)
cust_prfrd_lng VARCHAR(255)
cust_status VARCHAR(255)
cust_type VARCHAR(255)
cust_efctv_dt DATE
cust_host_ts DATETIME(6)
cust_local_ts DATETIME(6)
cust_user_id VARCHAR(255)
cust_uuid VARCHAR(255)
cust_ws_id VARCHAR(255)
Indexes

cust_nme_cmpnt

id BIGINT(20)
cstnmcmp_crud_value CHAR(1)
cstnmcmp_prgm_id VARCHAR(255)
cstnmcmp_acpt_ts DATETIME(6)
cstnmcmp_acpt_ts_utc_ofst DATETIME(6)
cstid_idfr BIGINT(20)
cstnmcmp_type_id VARCHAR(255)
cstnmcmp_type_value VARCHAR(255)
cstnmcmp_efctv_dt DATE
cstnmcmp_host_ts DATETIME(6)
cstnmcmp_local_ts DATETIME(6)
cstnmcmp_user_id VARCHAR(255)
cstnmcmp_uuid VARCHAR(255)
cstnmcmp_ws_id VARCHAR(255)
Indexes

cust_addr_cmpnt

id BIGINT(20)
cstadcmp_crud_value CHAR(1)
cstadcmp_prgm_id VARCHAR(255)
cstadcmp_acpt_ts DATETIME(6)
cstadcmp_acpt_ts_utc_ofst DATETIME...
cstadcmp_type_id VARCHAR(255)
cstadcmp_type_value VARCHAR(255)
cstadcmp_efctv_dt DATE
cstadcmp_host_ts DATETIME(6)
cstadcmp_local_ts DATETIME(6)
cstadcmp_user_id VARCHAR(255)
cstadcmp_uuid VARCHAR(255)
cstadcmp_ws_id VARCHAR(255)
cstid_idfr BIGINT(20)
Indexes

cust_dtl

cst_idfr BIGINT(20)
cust_crud_value CHAR(1)
cust_prgm_id VARCHAR(255)
cust_acpt_ts DATETIME(6)
cust_acpt_ts_utc_ofst DATETIME(6)
cust_cntry_orgtn VARCHAR(255)
cust_dob DATE
cust_gendr VARCHAR(255)
cstid_idfr BIGINT(20)
cust_prfrd_lng VARCHAR(255)
cust_status VARCHAR(255)
cust_type VARCHAR(255)
cust_efctv_dt DATE
cust_host_ts DATETIME(6)
cust_local_ts DATETIME(6)
cust_user_id VARCHAR(255)
cust_uuid VARCHAR(255)
cust_ws_id VARCHAR(255)
Indexes

cust_cl

cstcl_id BIGINT(20)
cstcl_crud_value CHAR(1)
cstcl_typ_value VARCHAR(255)
cstcl_prgm_id VARCHAR(255)
cstcl_acpt_ts DATETIME(6)
cstcl_acpt_ts_utc_ofst DATETIME(6)
cstcl_typ VARCHAR(255)
cstcl_efctv_dt DATE
cstcl_host_ts DATETIME(6)
cstcl_local_ts DATETIME(6)
cstcl_user_id VARCHAR(255)
cstcl_uuid VARCHAR(255)
cstcl_ws_id VARCHAR(255)
Indexes

execution_log

id BIGINT(20)
execution_time BIGINT(20)
method_name VARCHAR(255)
parameters LONGTEXT
result LONGTEXT
success BIT(1)
timestamp DATETIME(6)
Indexes

cust_idfn

cstin_idfr BIGINT(20)
cstin_crud_value CHAR(1)
cstin_prgm_id VARCHAR(255)
cstin_acpt_ts DATETIME(6)
cstin_acpt_ts_utc_ofst DATETIME(6)
cstin_id_item VARCHAR(255)
cstin_id_type VARCHAR(255)
cstin_efctv_dt DATE
cstin_host_ts DATETIME(6)
cstin_local_ts DATETIME(6)
cstin_user_id VARCHAR(255)
cstin_uuid VARCHAR(255)
cstin_ws_id VARCHAR(255)
Indexes

cstmr_signin

usr_idfr BIGINT(20)
usr_crud_value CHAR(1)
usr_prgm_id VARCHAR(255)
usr_acpt_ts DATETIME(6)
usr_acpt_ts_utc_ofst DATETIME(6)
usr_efctv_dt DATE
usr_host_ts DATETIME(6)
usr_local_ts DATETIME(6)
usr_password VARCHAR(255)
usr_user_id VARCHAR(255)
usr_usrname VARCHAR(255)
usr_uuid VARCHAR(255)
usr_ws_id VARCHAR(255)
cstid_idfr BIGINT(20)
Indexes

cust_pridn

id BIGINT(20)
cstprid_crud_value CHAR(1)
cstprid_prgm_id VARCHAR(255)
cstprid_acpt_ts DATETIME(6)
cstprid_acpt_ts_utc_ofst DATETIME(6)
cstprid_value VARCHAR(255)
cstprid_end_dt DATE
cstprid_host_ts DATETIME(6)
cstprid_local_ts DATETIME(6)
cstprid_strt_dt DATE
cstprid_user_id VARCHAR(255)
cstprid_uuid VARCHAR(255)
cstprid_ws_id VARCHAR(255)
cstcl_id BIGINT(20)
cstid_idfr BIGINT(20)
Indexes

cust_cntct_dtls

cstcndt_id BIGINT(20)
cstcndt_crud_value CHAR(1)
cstcndt_prgm_id VARCHAR(255)
cstcndt_acpt_ts DATETIME(6)
cstcndt_acpt_ts_utc_ofst DATETIME(6)
cstcndt_value VARCHAR(255)
cstcndt_end_dt DATE
cstcndt_host_ts DATETIME(6)
cstcndt_local_ts DATETIME(6)
cstcndt_strt_dt DATE
cstcndt_user_id VARCHAR(255)
cstcndt_uuid VARCHAR(255)
cstcndt_ws_id VARCHAR(255)
cstid_idfr BIGINT(20)
Indexes

Q2: Define the physical tables in the table format given as sample for Customer Classification with the data type, size and explanation of each field.

Sol:

Table Name: CUST_DTL (Customer Details)
Entity Name: Customer Details
Short Name: CUST_DTL
Description:
Stores core demographic and status information of customers. Acts as the master table for customer-related entities.
Attributes
- cst_idfr – BIGINT – Primary Key, uniquely identifies a customer
- cust_cntry_orgtn – VARCHAR(255) – Customer country of origin
- cust_dob – DATE – Customer date of birth
- cust_gendr – VARCHAR(255) – Customer gender
- cust_prfrd_lng – VARCHAR(255) – Preferred language
- cust_status – VARCHAR(255) – Customer status (Active/Inactive)
- cust_type – VARCHAR(255) – Customer type (Individual/Corporate)
- cust_efctv_dt – DATE – Date from which record is effective
- cust_crud_value – CHAR(1) – C-Created, U-Updated, D-Deleted
- cust_prgm_id – VARCHAR(255) – Program inserting the record
- cust_user_id – VARCHAR(255) – User who created the record
- cust_ws_id – VARCHAR(255) – Workstation ID
- cust_host_ts – DATETIME(6) – Host system timestamp
- cust_local_ts – DATETIME(6) – Local machine timestamp
- cust_acpt_ts – DATETIME(6) – Acceptance timestamp
- cust_acpt_ts_utc_ofst – DATETIME(6) – UTC offset timestamp
- cust_uuid – VARCHAR(255) – Universal unique identifier

Table Name: CUST_CL (Customer Classification)
Entity Name: Customer Classification
Short Name: CUST_CL
Description:

Stores classification types and values used across customer entities such as name type, address type, ID type, etc.
Attributes
- cstcl_id – BIGINT – Primary Key
- cstcl_typ – VARCHAR(255) – Classification type
- cstcl_typ_value – VARCHAR(255) – Classification value
- cstcl_efctv_dt – DATE – Effective date
- cstcl_crud_value – CHAR(1) – Record status (C/U/D)
- cstcl_prgm_id – VARCHAR(255) – Program ID
- cstcl_user_id – VARCHAR(255) – User ID
- cstcl_ws_id – VARCHAR(255) – Workstation ID
- cstcl_host_ts – DATETIME(6) – Host timestamp
- cstcl_local_ts – DATETIME(6) – Local timestamp
- cstcl_acpt_ts – DATETIME(6) – Acceptance timestamp
- cstcl_acpt_ts_utc_ofst – DATETIME(6) – UTC offset timestamp
- cstcl_uuid – VARCHAR(255) – Universal unique identifier

Table Name: CUST_NME_CMPNT (Customer Name Component)
Description:
Stores different name components of a customer such as first name, middle name, and last name.
Attributes
- id – BIGINT – Primary Key
- cstid_idfr – BIGINT – Foreign Key referencing CUST_DTL
- cstnmcmp_type_id – VARCHAR(255) – Name type identifier
- cstnmcmp_type_value – VARCHAR(255) – Name value
- cstnmcmp_efctv_dt – DATE – Effective date
- cstnmcmp_crud_value – CHAR(1) – C/U/D
- cstnmcmp_prgm_id – VARCHAR(255) – Program ID
- cstnmcmp_user_id – VARCHAR(255) – User ID
- cstnmcmp_ws_id – VARCHAR(255) – Workstation ID
- cstnmcmp_host_ts – DATETIME(6) – Host timestamp
- cstnmcmp_local_ts – DATETIME(6) – Local timestamp
- cstnmcmp_acpt_ts – DATETIME(6) – Acceptance timestamp
- cstnmcmp_acpt_ts_utc_ofst – DATETIME(6) – UTC offset timestamp
- cstnmcmp_uuid – VARCHAR(255) – Unique identifier

Table Name: CUST_ADDR_CMPNT (Customer Address Component)
Description:
Stores address components of customers such as address line, city, state, etc.
Attributes

- id – BIGINT – Primary Key
- cstid_idfr – BIGINT – Foreign Key referencing CUST_DTL
- cstadcmp_type_id – VARCHAR(255) – Address type ID
- cstadcmp_type_value – VARCHAR(255) – Address value
- cstadcmp_efctv_dt – DATE – Effective date
- cstadcmp_crud_value – CHAR(1) – C/U/D
- cstadcmp_prgm_id – VARCHAR(255) – Program ID
- cstadcmp_user_id – VARCHAR(255) – User ID
- cstadcmp_ws_id – VARCHAR(255) – Workstation ID
- cstadcmp_host_ts – DATETIME(6) – Host timestamp
- cstadcmp_local_ts – DATETIME(6) – Local timestamp
- cstadcmp_acpt_ts – DATETIME(6) – Acceptance timestamp
- cstadcmp_acpt_ts_utc_ofst – DATETIME(6) – UTC offset timestamp
- cstadcmp_uuid – VARCHAR(255) – Unique identifier

Table Name: CUST_CNTCT_DTLS (Customer Contact Details)
Description:
Stores customer contact information such as phone number and email.
Attributes

- cstcndt_id – BIGINT – Primary Key
- cstid_idfr – BIGINT – Foreign Key referencing CUST_DTL
- cstcndt_value – VARCHAR(255) – Contact value
- cstcndt_strt_dt – DATE – Start date
- cstcndt_end_dt – DATE – End date
- cstcndt_crud_value – CHAR(1) – C/U/D
- cstcndt_prgm_id – VARCHAR(255) – Program ID
- cstcndt_user_id – VARCHAR(255) – User ID
- cstcndt_ws_id – VARCHAR(255) – Workstation ID
- cstcndt_host_ts – DATETIME(6) – Host timestamp
- cstcndt_local_ts – DATETIME(6) – Local timestamp
- cstcndt_acpt_ts – DATETIME(6) – Acceptance timestamp
- cstcndt_acpt_ts_utc_ofst – DATETIME(6) – UTC offset timestamp

- cstcndt_uuid – VARCHAR(255) – Unique identifier

Table Name: CUST_IDFN (Customer Identification)
Description:
Stores identification details of customers such as Aadhaar, Passport, etc.
Attributes

- cstin_idfr – BIGINT – Primary Key
- cstid_idfr – BIGINT – Foreign Key referencing CUST_DTL
- cstin_id_type – VARCHAR(255) – Identification type
- cstin_id_item – VARCHAR(255) – Identification value
- cstin_efctv_dt – DATE – Effective date
- cstin_crud_value – CHAR(1) – C/U/D
- cstin_prgm_id – VARCHAR(255) – Program ID
- cstin_user_id – VARCHAR(255) – User ID
- cstin_ws_id – VARCHAR(255) – Workstation ID
- cstin_host_ts – DATETIME(6) – Host timestamp
- cstin_local_ts – DATETIME(6) – Local timestamp
- cstin_acpt_ts – DATETIME(6) – Acceptance timestamp
- cstin_acpt_ts_utc_ofst – DATETIME(6) – UTC offset timestamp
- cstin_uuid – VARCHAR(255) – Unique identifier

Table Name: CUST_PRIDN (Customer Proof Identification)
Description:
Stores proof identification details linked to customer classification.
Attributes

- id – BIGINT – Primary Key
- cstid_idfr – BIGINT – Foreign Key referencing CUST_DTL
- cstcl_id – BIGINT – Foreign Key referencing CUST_CL
- cstprid_value – VARCHAR(255) – Proof value
- cstprid_strt_dt – DATE – Start date
- cstprid_end_dt – DATE – End date
- cstprid_crud_value – CHAR(1) – C/U/D
- cstprid_prgm_id – VARCHAR(255) – Program ID
- cstprid_user_id – VARCHAR(255) – User ID
- cstprid_ws_id – VARCHAR(255) – Workstation ID
- cstprid_host_ts – DATETIME(6) – Host timestamp

- cstprid_local_ts – DATETIME(6) – Local timestamp
- cstprid_acpt_ts – DATETIME(6) – Acceptance timestamp
- cstprid_acpt_ts_utc_ofst – DATETIME(6) – UTC offset timestamp
- cstprid_uuid – VARCHAR(255) – Unique identifier

Table Name: CSTMR_SIGNIN (Customer Sign-In)
Description:
Stores authentication and login details of customers.
Attributes

- usr_idfr – BIGINT – Primary Key
- usr_usrname – VARCHAR(255) – Username
- usr_password – VARCHAR(255) – Encrypted password
- cstid_idfr – BIGINT – Foreign Key referencing CUST_DTL
- usr_efctv_dt – DATE – Effective date
- usr_crud_value – CHAR(1) – C/U/D
- usr_prgm_id – VARCHAR(255) – Program ID
- usr_user_id – VARCHAR(255) – User ID
- usr_ws_id – VARCHAR(255) – Workstation ID
- usr_host_ts – DATETIME(6) – Host timestamp
- usr_local_ts – DATETIME(6) – Local timestamp
- usr_acpt_ts – DATETIME(6) – Acceptance timestamp
- usr_acpt_ts_utc_ofst – DATETIME(6) – UTC offset timestamp
- usr_uuid – VARCHAR(255) – Unique identifier

Table Name: EXECUTION_LOG
Description:
Stores execution logs for system auditing and debugging.
Attributes

- id – BIGINT – Primary Key
- execution_time – BIGINT – Execution time in milliseconds
- method_name – VARCHAR(255) – Method executed
- parameters – LONGTEXT – Input parameters
- result – LONGTEXT – Execution result
- success – BIT(1) – Execution status
- timestamp – DATETIME(6) – Execution timestamp

Q3: Create Schema and tables SQL’s, also create the DDL scripts for table creation.

Sol:

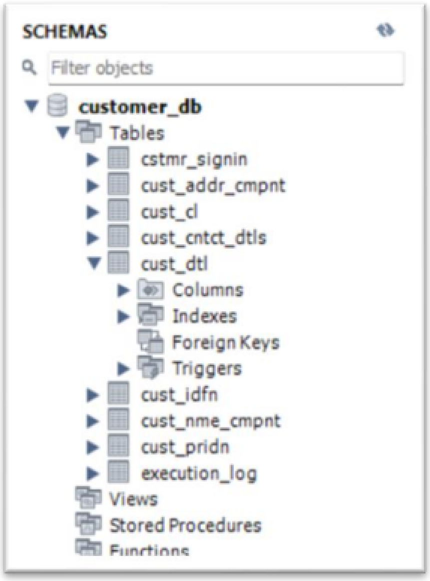
Step 1: Create Schema

```
CREATE SCHEMA CUSTOMER_DB;
USE CUSTOMER_DB;
```

Step 2: Table Creation Scripts (DDL)

```
CREATE TABLE CUST_DTL (
 cst_idfr BIGINT PRIMARY KEY,
 cust_cntry_orgtn VARCHAR(255),
 cust_dob DATE,
 cust_gendr VARCHAR(255),
 cust_prfrd_lng VARCHAR(255),
 cust_status VARCHAR(255),
 cust_type VARCHAR(255),
 cust_efctv_dt DATE,
 cust_crud_value CHAR(1),
 cust_prgm_id VARCHAR(255),
 cust_user_id VARCHAR(255),
 cust_ws_id VARCHAR(255),
 cust_host_ts DATETIME(6),
 cust_local_ts DATETIME(6),
 cust_acpt_ts DATETIME(6),
 cust_acpt_ts_utc_ofst DATETIME(6),
 cust_uuid VARCHAR(255)
);
CREATE TABLE CUST_CL (
 cstcl_id BIGINT PRIMARY KEY,
 cstcl_typ VARCHAR(255),
 cstcl_typ_value VARCHAR(255),
 cstcl_efctv_dt DATE,
 cstcl_crud_value CHAR(1),
 cstcl_prgm_id VARCHAR(255),
 cstcl_user_id VARCHAR(255),
 cstcl_ws_id VARCHAR(255),
 cstcl_host_ts DATETIME(6),
 cstcl_local_ts DATETIME(6),
 cstcl_acpt_ts DATETIME(6),
 cstcl_acpt_ts_utc_ofst DATETIME(6),
 cstcl_uuid VARCHAR(255)
);
CREATE TABLE CUST_NME_CMPNT (
 id BIGINT PRIMARY KEY,
 cstid_idfr BIGINT,
 cstnmcmp_type_id VARCHAR(255),
 cstnmcmp_type_value VARCHAR(255),
 cstnmcmp_efctv_dt DATE,
 cstnmcmp_crud_value CHAR(1),
 cstnmcmp_prgm_id VARCHAR(255),
 cstnmcmp_user_id VARCHAR(255),
 cstnmcmp_ws_id VARCHAR(255),
 cstnmcmp_host_ts DATETIME(6),
 cstnmcmp_local_ts DATETIME(6),
 cstnmcmp_acpt_ts DATETIME(6),
 cstnmcmp_acpt_ts_utc_ofst DATETIME(6),
cstnmcmp_uuid VARCHAR(255),
 CONSTRAINT fk_nme_customer
 FOREIGN KEY (cstid_idfr)
 REFERENCES CUST_DTL(cst_idfr)
);
CREATE TABLE CUST_ADDR_CMPNT (
 id BIGINT PRIMARY KEY,
 cstid_idfr BIGINT,
 cstadcmp_type_id VARCHAR(255),
 cstadcmp_type_value VARCHAR(255),
 cstadcmp_efctv_dt DATE,
 cstadcmp_crud_value CHAR(1),
 cstadcmp_prgm_id VARCHAR(255),
 cstadcmp_user_id VARCHAR(255),
 cstadcmp_ws_id VARCHAR(255),
 cstadcmp_host_ts DATETIME(6),
 cstadcmp_local_ts DATETIME(6),
 cstadcmp_acpt_ts DATETIME(6),
 cstadcmp_acpt_ts_utc_ofst DATETIME(6),
 cstadcmp_uuid VARCHAR(255),
 CONSTRAINT fk_addr_customer
 FOREIGN KEY (cstid_idfr)
 REFERENCES CUST_DTL(cst_idfr)
);
CREATE TABLE CUST_CNTCT_DTLS (
 cstcndt_id BIGINT PRIMARY KEY,
 cstid_idfr BIGINT,
 cstcndt_value VARCHAR(255),
 cstcndt_strt_dt DATE,
cstcndt_end_dt DATE,
 cstcndt_crud_value CHAR(1),
 cstcndt_prgm_id VARCHAR(255),
 cstcndt_user_id VARCHAR(255),
 cstcndt_ws_id VARCHAR(255),
 cstcndt_host_ts DATETIME(6),
 cstcndt_local_ts DATETIME(6),
 cstcndt_acpt_ts DATETIME(6),
 cstcndt_acpt_ts_utc_ofst DATETIME(6),
 cstcndt_uuid VARCHAR(255),
 CONSTRAINT fk_cntct_customer
 FOREIGN KEY (cstid_idfr)
 REFERENCES CUST_DTL(cst_idfr)
);
CREATE TABLE CUST_IDFN (
 cstin_idfr BIGINT PRIMARY KEY,
 cstid_idfr BIGINT,
 cstin_id_type VARCHAR(255),
 cstin_id_item VARCHAR(255),
 cstin_efctv_dt DATE,
 cstin_crud_value CHAR(1),
 cstin_prgm_id VARCHAR(255),
 cstin_user_id VARCHAR(255),
 cstin_ws_id VARCHAR(255),
 cstin_host_ts DATETIME(6),
 cstin_local_ts DATETIME(6),
 cstin_acpt_ts DATETIME(6),
 cstin_acpt_ts_utc_ofst DATETIME(6),
 cstin_uuid VARCHAR(255),
CONSTRAINT fk_idfn_customer
 FOREIGN KEY (cstid_idfr)
 REFERENCES CUST_DTL(cst_idfr)
);
CREATE TABLE CUST_PRIDN (
 id BIGINT PRIMARY KEY,
 cstid_idfr BIGINT,
 cstcl_id BIGINT,
 cstprid_value VARCHAR(255),
 cstprid_strt_dt DATE,
 cstprid_end_dt DATE,
 cstprid_crud_value CHAR(1),
 cstprid_prgm_id VARCHAR(255),
 cstprid_user_id VARCHAR(255),
 cstprid_ws_id VARCHAR(255),
 cstprid_host_ts DATETIME(6),
 cstprid_local_ts DATETIME(6),
 cstprid_acpt_ts DATETIME(6),
 cstprid_acpt_ts_utc_ofst DATETIME(6),
 cstprid_uuid VARCHAR(255),
 CONSTRAINT fk_pridn_customer
 FOREIGN KEY (cstid_idfr)
 REFERENCES CUST_DTL(cst_idfr),
 CONSTRAINT fk_pridn_classification
 FOREIGN KEY (cstcl_id)
 REFERENCES CUST_CL(cstcl_id)
);
CREATE TABLE CSTMR_SIGNIN (
 usr_idfr BIGINT PRIMARY KEY,
usr_usrname VARCHAR(255),
 usr_password VARCHAR(255),
 usr_efctv_dt DATE,
 usr_crud_value CHAR(1),
 usr_prgm_id VARCHAR(255),
 usr_user_id VARCHAR(255),
 usr_ws_id VARCHAR(255),
 usr_host_ts DATETIME(6),
 usr_local_ts DATETIME(6),
 usr_acpt_ts DATETIME(6),
 usr_acpt_ts_utc_ofst DATETIME(6),
 usr_uuid VARCHAR(255),
 cstid_idfr BIGINT,
 CONSTRAINT fk_signin_customer
 FOREIGN KEY (cstid_idfr)
 REFERENCES CUST_DTL(cst_idfr)
);
CREATE TABLE EXECUTION_LOG (
 id BIGINT PRIMARY KEY,
 execution_time BIGINT,
 method_name VARCHAR(255),
 parameters LONGTEXT,
 result LONGTEXT,
 success BIT(1),
 timestamp DATETIME(6)
);
```

All tables were created using scripts in MySQL Workbench. Primary and foreign key constraints implemented as per ER diagram.



```
SCHEMAS
Filter objects
▼ customer_db
  ▼ Tables
    ▶ cstmr_signin
    ▶ cust_addr_cmpnt
    ▶ cust_cl
    ▶ cust_cntct_dtls
    ▼ cust_dtl
      ▶ Columns
      ▶ Indexes
        Foreign Keys
      ▶ Triggers
    ▶ cust_idfn
    ▶ cust_nme_cmpnt
    ▶ cust_pridn
    ▶ execution_log
  Views
  Stored Procedures
  Functions
```

Q4: DML Operations on table created - SELECT, UPDATE and DELETE.

Sol:

Step 1: INSERT Operation:
Insert data into CUST_DTL

```
INSERT INTO CUST_DTL (
 cst_idfr,
 cust_cntry_orgtn,
 cust_dob,
 cust_gendr,
 cust_prfrd_lng,
 cust_status,
 cust_type,
 cust_efctv_dt,
 cust_crud_value,
 cust_prgm_id,
 cust_user_id,
 cust_ws_id,
 cust_host_ts,
 cust_local_ts,
 cust_acpt_ts,
 cust_acpt_ts_utc_ofst,
 cust_uuid
)
VALUES (
 1001,
 'India',
 '2003-05-12',
'Male',
 'English',
 'Active',
 'Individual',
 CURDATE(),
 'C',
 'CUST_APP',
 'admin',
 'WS01',
 CURRENT_TIMESTAMP,
 CURRENT_TIMESTAMP,
 CURRENT_TIMESTAMP,
 CURRENT_TIMESTAMP,
 'UUID-1001'
);
```

Insert data into CUST_CL

```
INSERT INTO CUST_CL (
 cstcl_id,
 cstcl_typ,
 cstcl_typ_value,
 cstcl_efctv_dt,
 cstcl_crud_value,
 cstcl_prgm_id,
 cstcl_user_id,
 cstcl_ws_id,
 cstcl_host_ts,
 cstcl_local_ts,
 cstcl_acpt_ts,
 Cstcl_acpt_ts_utc_ofst,
cstcl_uuid
)
VALUES (
 1,
 'ID Type',
 'Aadhaar',
 CURDATE(),
 'C',
 'CUST_APP',
 'admin',
 'WS01',
 CURRENT_TIMESTAMP,
 CURRENT_TIMESTAMP,
 CURRENT_TIMESTAMP,
 CURRENT_TIMESTAMP,
 'UUID-CL-1'
);
```

Step 2: SELECT Operation

Select all customers

```
SELECT * FROM CUST_DTL;
```

Step 3: UPDATE Operation

Update customer status

```
UPDATE CUST_DTL
SET cust_status = 'Inactive',
 cust_crud_value = 'U'
WHERE cst_idfr = 1001;
```

Step 4: DELETE Operation

```
UPDATE CUST_DTL
SET cust_crud_value = 'D'
WHERE cst_idfr = 1001;
```