

CS-GY 6083

Principles of Database Systems

Section B

NEW YORK UNIVERSITY
Tandon School of Engineering
Spring 2024

Group Project – Part 1 Submission

GROUP MEMBERS

Durga Avinash Kodavalla || dk4852 || N11128244

Chirag Chopra || cc7083 || N18722294

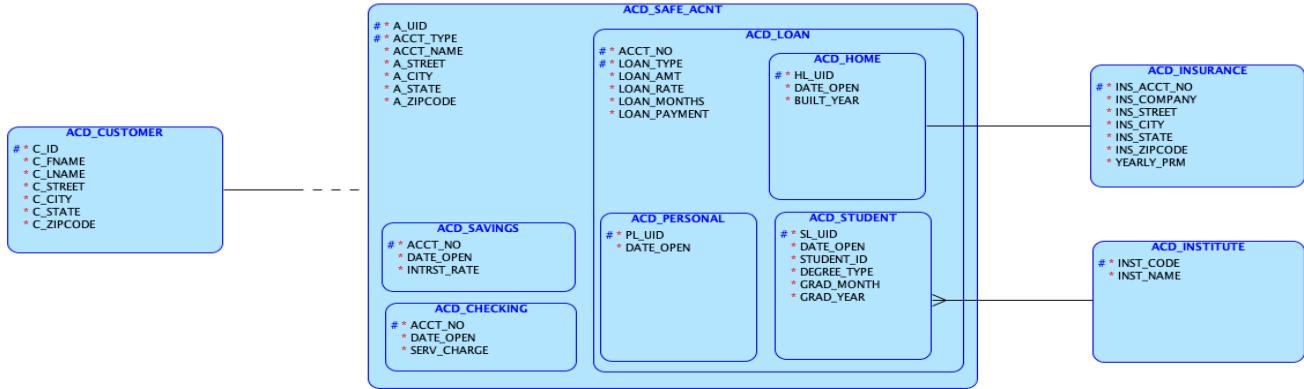
Arushi Ojha || ao2618 || N14118398

Date of Submission
03/28/2024

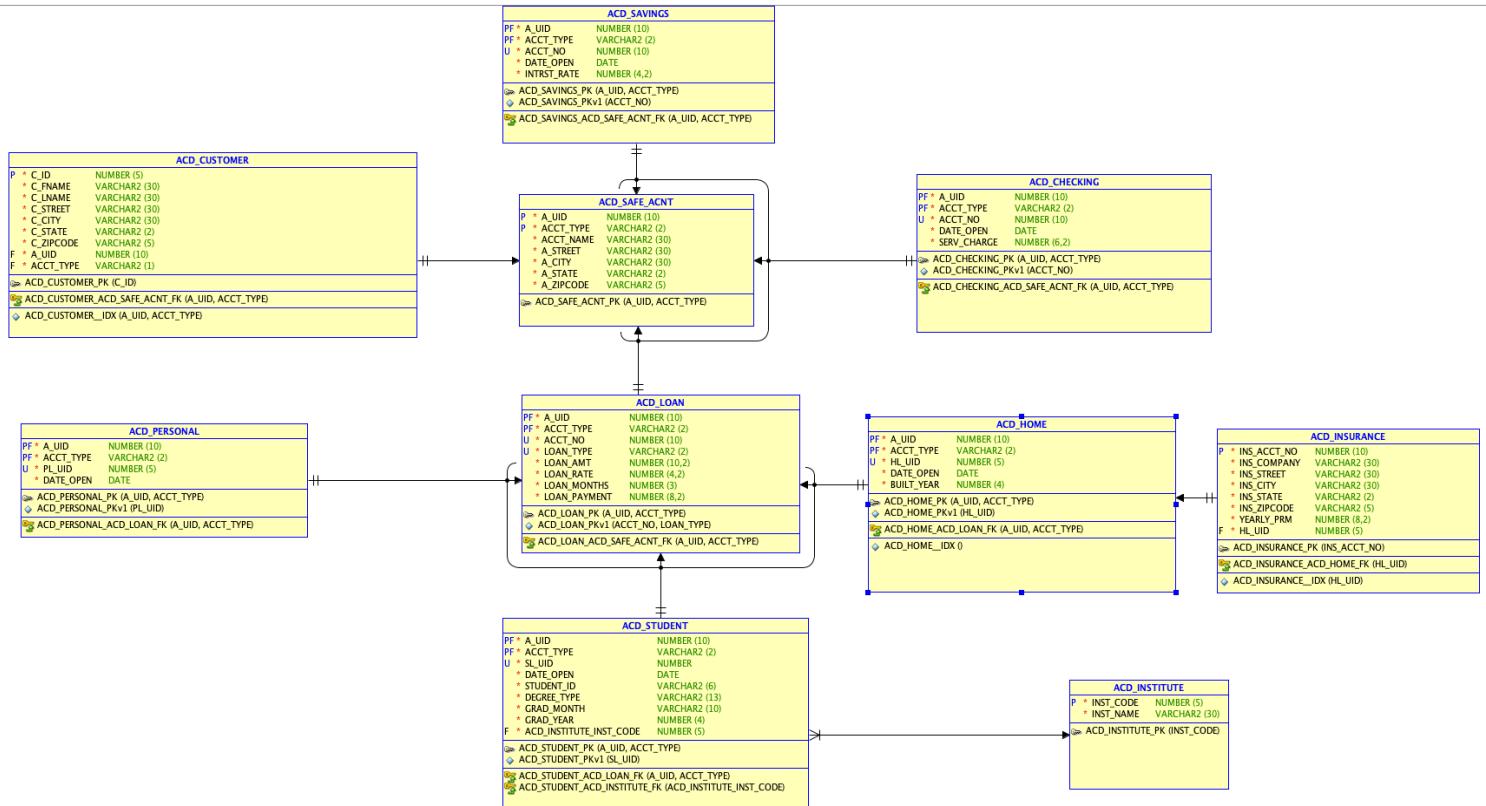
TABLE OF CONTENTS

Section Name	Page Number
1. Logical E-R Model	3
2. Relational Model	3
3. Assumptions	4
4. DDL Code as Generated from Relational Model	5
5. DDL Code After Applying CHECK Constraint(s)	18
6. Data Population DML Code	31
7. Count Of Records from Each Entity	38
8. Data Dictionary Queries Detailing All Tables	42
9. MySQL Code – Database	46

1. Logical E-R Model



2. Relational Model



3. Assumptions

- A customer will have unique account ID (A_UID in ACD_SAFE_ACNT) and will have different account numbers for different types of accounts in it.
- A customer can also have multiple loan types (student/home/personal) at once in a single loan account with their own unique combination of account number and loan type.
- Each type of loan has their unique ID (HL_UID, SL_UID, PL_UID).
- A_UID & ACNT_TYPE in ACD_SAFE_ACNT and ACCT_NO & LOAN_TYPE in ACD_LOAN is a pair primary key to have one/many subtypes condition.
- For home loans, it is assumed to have an insurance (only one & mandatory).
- Each customer in ACD_CUSTOMER is assigned with their unique ID.
- Insurance account number (INS_ACCT_NO in ACD_INSURANCE) is assumed to be unique irrespective of the insurance companies available in the market.
- Customer closed bank account data are also stored with 1 – 1 (optional) relationship with customer and safe account. Its kept optional to consider the case where customer has closed the account in the bank, but the data is still there with the bank.
- Applied a check constraint for undergraduate and graduate type of degree.
- In future if some change comes for adding an attribute for personal loan, we have kept that entity in Loan supertype and kept one attribute as Date Opened at an individual level for all the entities.

4. DDL Code as Generated from Relational Model

```
-- Generated by Oracle SQL Developer Data Modeler 23.1.0.087.0806
-- at: 2024-03-28 18:37:14 EDT
-- site: Oracle Database 21c
-- type: Oracle Database 21c

-- predefined type, no DDL - MDSYS.SDO_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE acd_checking (
    a_uid      NUMBER(10) NOT NULL,
    acct_type  VARCHAR2(2) NOT NULL,
    acct_no    NUMBER(10) NOT NULL,
    date_open  DATE NOT NULL,
    serv_charge NUMBER(6, 2) NOT NULL
);

COMMENT ON COLUMN acd_checking.a_uid IS
    'SAFE Unique Account ID';

COMMENT ON COLUMN acd_checking.acct_type IS
    'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_checking.acct_no IS
    'Checking Account Unique Account Number';

COMMENT ON COLUMN acd_checking.date_open IS
    'Account Opening Date';

COMMENT ON COLUMN acd_checking.serv_charge IS
    'Account Service Charge';

ALTER TABLE acd_checking ADD CONSTRAINT acd_checking_pk PRIMARY KEY ( a_uid,
                                                               acct_type );

ALTER TABLE acd_checking ADD CONSTRAINT acd_checking_pkv1 UNIQUE ( acct_no );

CREATE TABLE acd_customer (
    c_id      NUMBER(5) NOT NULL,
    c_fname   VARCHAR2(30) NOT NULL,
    c_lname   VARCHAR2(30) NOT NULL,
    c_street  VARCHAR2(30) NOT NULL,
    c_city    VARCHAR2(30) NOT NULL,
    c_state   VARCHAR2(2) NOT NULL,
    c_zipcode VARCHAR2(5) NOT NULL,
    a_uid     NUMBER(10) NOT NULL,
    acct_type VARCHAR2(1) NOT NULL
```

```

);

COMMENT ON COLUMN acd_customer.c_id IS
    'Unique Customer ID';

COMMENT ON COLUMN acd_customer.c_fname IS
    'Customer First Name';

COMMENT ON COLUMN acd_customer.c_lname IS
    'Customer Last Name';

COMMENT ON COLUMN acd_customer.c_street IS
    'Customer Street – Address Line 1';

COMMENT ON COLUMN acd_customer.c_city IS
    'Customer City Location';

COMMENT ON COLUMN acd_customer.c_state IS
    'Customer State Code – Ex: New York: NY';

COMMENT ON COLUMN acd_customer.c_zipcode IS
    'Customer Zipcode';

CREATE UNIQUE INDEX acd_customer_idx ON
    acd_customer (
        a_uid
    ASC,
        acct_type
    ASC );

ALTER TABLE acd_customer ADD CONSTRAINT acd_customer_pk PRIMARY KEY ( c_id );

CREATE TABLE acd_home (
    a_uid      NUMBER(10) NOT NULL,
    acct_type  VARCHAR2(2) NOT NULL,
    hl_uid     NUMBER(5) NOT NULL,
    date_open   DATE NOT NULL,
    built_year NUMBER(4) NOT NULL
);

COMMENT ON COLUMN acd_home.a_uid IS
    'SAFE Unique Account ID';

COMMENT ON COLUMN acd_home.acct_type IS
    'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_home.hl_uid IS
    'Home Loan Unique Loan ID';

COMMENT ON COLUMN acd_home.date_open IS
    'Home Loan Account Opening Date';

```

```

COMMENT ON COLUMN acd_home.built_year IS
    'Home Built Year';

ALTER TABLE acd_home ADD CONSTRAINT acd_home_pk PRIMARY KEY ( a_uid,
                                                               acct_type );

ALTER TABLE acd_home ADD CONSTRAINT acd_home_pkv1 UNIQUE ( hl_uid );

CREATE TABLE acd_institute (
    inst_code NUMBER(5) NOT NULL,
    inst_name VARCHAR2(30) NOT NULL
);

COMMENT ON COLUMN acd_institute.inst_code IS
    'Unique Institute Code';

COMMENT ON COLUMN acd_institute.inst_name IS
    'Institute Name (University Name)';

ALTER TABLE acd_institute ADD CONSTRAINT acd_institute_pk PRIMARY KEY ( inst_code
);

CREATE TABLE acd_insurance (
    ins_acct_no NUMBER(10) NOT NULL,
    ins_company VARCHAR2(30) NOT NULL,
    ins_street   VARCHAR2(30) NOT NULL,
    ins_city     VARCHAR2(30) NOT NULL,
    ins_state    VARCHAR2(2) NOT NULL,
    ins_zipcode  VARCHAR2(5) NOT NULL,
    yearly_prm  NUMBER(8, 2) NOT NULL,
    hl_uid       NUMBER(5) NOT NULL
);

COMMENT ON COLUMN acd_insurance.ins_acct_no IS
    'Home Insurance Unique Account Number';

COMMENT ON COLUMN acd_insurance.ins_company IS
    'Insurance Company Name';

COMMENT ON COLUMN acd_insurance.ins_street IS
    'Insurance Street Address';

COMMENT ON COLUMN acd_insurance.ins_city IS
    'Insurance City Location';

COMMENT ON COLUMN acd_insurance.ins_state IS
    'Insurance State Code- Ex: New York: NY';

COMMENT ON COLUMN acd_insurance.ins_zipcode IS

```

```

'Insurance Company''s Zipcode';

COMMENT ON COLUMN acd_insurance.yearly_prm IS
  'Insurance Yearly Premium';

CREATE UNIQUE INDEX acd_insurance__idx ON
  acd_insurance (
    hl_uid
  ASC );

ALTER TABLE acd_insurance ADD CONSTRAINT acd_insurance_pk PRIMARY KEY ( ins_acct_no
);

CREATE TABLE acd_loan (
  a_uid      NUMBER(10) NOT NULL,
  acct_type  VARCHAR2(2) NOT NULL,
  acct_no    NUMBER(10) NOT NULL,
  loan_type  VARCHAR2(2) NOT NULL,
  loan_amt   NUMBER(10, 2) NOT NULL,
  loan_rate  NUMBER(4, 2) NOT NULL,
  loan_months NUMBER(3) NOT NULL,
  loan_payment NUMBER(8, 2) NOT NULL
);

ALTER TABLE acd_loan
  ADD CONSTRAINT ch_inh_acd_loan CHECK ( loan_type IN ( 'HL', 'PL', 'SL' ) );

COMMENT ON COLUMN acd_loan.a_uid IS
  'SAFE Unique Account ID';

COMMENT ON COLUMN acd_loan.acct_type IS
  'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_loan.acct_no IS
  'Loan Account Unique Account Number';

COMMENT ON COLUMN acd_loan.loan_type IS
  'Loan Type – Student (SL)/ Home (HL)/ Personal (PL)';

COMMENT ON COLUMN acd_loan.loan_amt IS
  'OVERALL LOAN AMOUNT';

COMMENT ON COLUMN acd_loan.loan_rate IS
  'LOAN INTEREST RATE';

COMMENT ON COLUMN acd_loan.loan_months IS
  'LOAN DURATION';

COMMENT ON COLUMN acd_loan.loan_payment IS
  'LOAN REPAYMENT PER MONTH';

```

```

ALTER TABLE acd_loan ADD CONSTRAINT acd_loan_pk PRIMARY KEY ( a_uid,
                                                               acct_type );

ALTER TABLE acd_loan ADD CONSTRAINT acd_loan_pkv1 UNIQUE ( acct_no,
                                                               loan_type );

CREATE TABLE acd_personal (
    a_uid      NUMBER(10) NOT NULL,
    acct_type  VARCHAR2(2) NOT NULL,
    pl_uid     NUMBER(5) NOT NULL,
    date_open  DATE NOT NULL
);

COMMENT ON COLUMN acd_personal.a_uid IS
    'SAFE Unique Account ID';

COMMENT ON COLUMN acd_personal.acct_type IS
    'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_personal.pl_uid IS
    'Personal Loan Unique ID';

COMMENT ON COLUMN acd_personal.date_open IS
    'Personal Loan Account Opening Date';

ALTER TABLE acd_personal ADD CONSTRAINT acd_personal_pk PRIMARY KEY ( a_uid,
                                                               acct_type );

ALTER TABLE acd_personal ADD CONSTRAINT acd_personal_pkv1 UNIQUE ( pl_uid );

CREATE TABLE acd_safe_acnt (
    a_uid      NUMBER(10) NOT NULL,
    acct_type  VARCHAR2(2) NOT NULL,
    acct_name  VARCHAR2(30) NOT NULL,
    a_street   VARCHAR2(30) NOT NULL,
    a_city     VARCHAR2(30) NOT NULL,
    a_state    VARCHAR2(2) NOT NULL,
    a_zipcode  VARCHAR2(5) NOT NULL
);

ALTER TABLE acd_safe_acnt
    ADD CONSTRAINT ch_inh_acd_safe_acnt CHECK ( acct_type IN ( 'C', 'HL', 'PL',
'S', 'SL', 'L' ) );

COMMENT ON COLUMN acd_safe_acnt.a_uid IS
    'SAFE Unique Account ID';

COMMENT ON COLUMN acd_safe_acnt.acct_type IS
    'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

```

```

COMMENT ON COLUMN acd_safe_acnt.acct_name IS
    'Account Name';

COMMENT ON COLUMN acd_safe_acnt.a_street IS
    'Account Street Address';

COMMENT ON COLUMN acd_safe_acnt.a_city IS
    'Account City Location';

COMMENT ON COLUMN acd_safe_acnt.a_state IS
    'Account State Code - Ex: New York: NY';

COMMENT ON COLUMN acd_safe_acnt.a_zipcode IS
    'Account Zipcode';

ALTER TABLE acd_safe_acnt ADD CONSTRAINT acd_safe_acnt_pk PRIMARY KEY ( a_uid,
    acct_type
);

CREATE TABLE acd_savings (
    a_uid      NUMBER(10) NOT NULL,
    acct_type  VARCHAR2(2) NOT NULL,
    acct_no    NUMBER(10) NOT NULL,
    date_open  DATE NOT NULL,
    intrst_rate NUMBER(4, 2) NOT NULL
);

COMMENT ON COLUMN acd_savings.a_uid IS
    'SAFE Unique Account ID';

COMMENT ON COLUMN acd_savings.acct_type IS
    'Account Type - Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_savings.acct_no IS
    'Savings Account Unique Account Number';

COMMENT ON COLUMN acd_savings.date_open IS
    'Savings Account Opening Date';

COMMENT ON COLUMN acd_savings.intrst_rate IS
    'Interest Rate';

ALTER TABLE acd_savings ADD CONSTRAINT acd_savings_pk PRIMARY KEY ( a_uid,
    acct_type );

ALTER TABLE acd_savings ADD CONSTRAINT acd_savings_pkv1 UNIQUE ( acct_no );

CREATE TABLE acd_student (
    a_uid      NUMBER(10) NOT NULL,
    acct_type  VARCHAR2(2) NOT NULL,
    sl_uid     NUMBER NOT NULL,

```

```

date_open    DATE NOT NULL,
student_id   VARCHAR2(6) NOT NULL,
degree_type  VARCHAR2(13) NOT NULL,
grad_month   VARCHAR2(10) NOT NULL,
grad_year    NUMBER(4) NOT NULL,
inst_code    NUMBER(5) NOT NULL
);

COMMENT ON COLUMN acd_student.a_uid IS
'SAFE Unique Account ID';

COMMENT ON COLUMN acd_student.acct_type IS
'Account Type - Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_student.sl_uid IS
'Student Loan Unique Loan ID';

COMMENT ON COLUMN acd_student.date_open IS
'Student Loan Account Opening Date';

COMMENT ON COLUMN acd_student.student_id IS
'Student ID From Institute';

COMMENT ON COLUMN acd_student.degree_type IS
'Degree Type - Graduate / Undergraduate';

COMMENT ON COLUMN acd_student.grad_month IS
'Graduation Month';

COMMENT ON COLUMN acd_student.grad_year IS
'Graduation Year';

ALTER TABLE acd_student ADD CONSTRAINT acd_student_pk PRIMARY KEY ( a_uid,
                                                               acct_type );

ALTER TABLE acd_student ADD CONSTRAINT acd_student_pkv1 UNIQUE ( sl_uid );

ALTER TABLE acd_checking
ADD CONSTRAINT acd_checking_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
                                                               acct_type )
REFERENCES acd_safe_acnt ( a_uid,
                           acct_type );

ALTER TABLE acd_customer
ADD CONSTRAINT acd_customer_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
                                                               acct_type )
REFERENCES acd_safe_acnt ( a_uid,
                           acct_type );

```

```

ALTER TABLE acd_home
  ADD CONSTRAINT acd_home_acd_loan_fk FOREIGN KEY ( a_uid,
                                                    acct_type )
    REFERENCES acd_loan ( a_uid,
                          acct_type );

ALTER TABLE acd_insurance
  ADD CONSTRAINT acd_insurance_acd_home_fk FOREIGN KEY ( hl_uid )
    REFERENCES acd_home ( hl_uid );

ALTER TABLE acd_loan
  ADD CONSTRAINT acd_loan_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
                                                       acct_type )
    REFERENCES acd_safe_acnt ( a_uid,
                               acct_type );

ALTER TABLE acd_personal
  ADD CONSTRAINT acd_personal_acd_loan_fk FOREIGN KEY ( a_uid,
                                                       acct_type )
    REFERENCES acd_loan ( a_uid,
                           acct_type );

ALTER TABLE acd_savings
  ADD CONSTRAINT acd_savings_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
                                                       acct_type )
    REFERENCES acd_safe_acnt ( a_uid,
                               acct_type );

ALTER TABLE acd_student
  ADD CONSTRAINT acd_student_acd_institute_fk FOREIGN KEY ( inst_code )
    REFERENCES acd_institute ( inst_code );

ALTER TABLE acd_student
  ADD CONSTRAINT acd_student_acd_loan_fk FOREIGN KEY ( a_uid,
                                                       acct_type )
    REFERENCES acd_loan ( a_uid,
                           acct_type );

CREATE OR REPLACE TRIGGER arc_fkarc_7_acd_personal BEFORE
  INSERT OR UPDATE OF a_uid, acct_type ON acd_personal
  FOR EACH ROW
DECLARE
  d VARCHAR2(2);
BEGIN
  SELECT
    a.loan_type
  INTO d
  FROM
    acd_loan a
  WHERE
    a.a_uid = :new.a_uid

```

```

        AND a.acct_type = :new.acct_type;

        IF ( d IS NULL OR d <> 'PL' ) THEN
            raise_application_error(-20223, 'FK ACD_PERSONAL_ACD_LOAN_FK in Table
ACD_PERSONAL violates Arc constraint on Table ACD_LOAN - discriminator column
LOAN_TYPE doesn''t have value ''PL'''
        );
    END IF;

EXCEPTION
    WHEN no_data_found THEN
        NULL;
    WHEN OTHERS THEN
        RAISE;
END;
/

CREATE OR REPLACE TRIGGER arc_fkarc_7_acd_student BEFORE
    INSERT OR UPDATE OF a_uid, acct_type ON acd_student
    FOR EACH ROW
DECLARE
    d VARCHAR2(2);
BEGIN
    SELECT
        a.loan_type
    INTO d
    FROM
        acd_loan a
    WHERE
        a.a_uid = :new.a_uid
        AND a.acct_type = :new.acct_type;

    IF ( d IS NULL OR d <> 'SL' ) THEN
        raise_application_error(-20223, 'FK ACD_STUDENT_ACD_LOAN_FK in Table
ACD_STUDENT violates Arc constraint on Table ACD_LOAN - discriminator column
LOAN_TYPE doesn''t have value ''SL'''
    );
    END IF;

EXCEPTION
    WHEN no_data_found THEN
        NULL;
    WHEN OTHERS THEN
        RAISE;
END;
/

CREATE OR REPLACE TRIGGER arc_fkarc_7_acd_home BEFORE
    INSERT OR UPDATE OF a_uid, acct_type ON acd_home
    FOR EACH ROW
DECLARE

```

```

d VARCHAR2(2);
BEGIN
  SELECT
    a.loan_type
  INTO d
  FROM
    acd_loan a
  WHERE
    a.a_uid = :new.a_uid
    AND a.acct_type = :new.acct_type;

  IF ( d IS NULL OR d <> 'HL' ) THEN
    raise_application_error(-20223, 'FK ACD_HOME_ACD_LOAN_FK in Table ACD_HOME
violates Arc constraint on Table ACD_LOAN – discriminator column LOAN_TYPE doesn''t
have value ''HL'''
  );
END IF;

EXCEPTION
  WHEN no_data_found THEN
    NULL;
  WHEN OTHERS THEN
    RAISE;
END;
/

CREATE OR REPLACE TRIGGER arc_fkarc_8_acd_loan BEFORE
  INSERT OR UPDATE OF a_uid, acct_type ON acd_loan
  FOR EACH ROW
DECLARE
  d VARCHAR2(2);
BEGIN
  SELECT
    a.acct_type
  INTO d
  FROM
    acd_safe_acnt a
  WHERE
    a.a_uid = :new.a_uid
    AND a.acct_type = :new.acct_type;

  IF ( d IS NULL OR d <> 'L' ) THEN
    raise_application_error(-20223, 'FK ACD_LOAN_ACD_SAFE_ACNT_FK in Table
ACD_LOAN violates Arc constraint on Table ACD_SAFE_ACNT – discriminator column
ACCT_TYPE doesn''t have value ''L'''
  );
END IF;

EXCEPTION
  WHEN no_data_found THEN
    NULL;

```

```

WHEN OTHERS THEN
    RAISE;
END;
/

CREATE OR REPLACE TRIGGER arc_fkarc_8_acd_savings BEFORE
    INSERT OR UPDATE OF a_uid, acct_type ON acd_savings
    FOR EACH ROW
DECLARE
    d VARCHAR2(2);
BEGIN
    SELECT
        a.acct_type
    INTO d
    FROM
        acd_safe_acnt a
    WHERE
        a.a_uid = :new.a_uid
        AND a.acct_type = :new.acct_type;

    IF ( d IS NULL OR d <> 'S' ) THEN
        raise_application_error(-20223, 'FK ACD_SAVINGS_ACD_SAFE_ACNT_FK in Table
ACD_SAVINGS violates Arc constraint on Table ACD_SAFE_ACNT - discriminator column
ACCT_TYPE doesn''t have value ''S'''');
    END IF;

EXCEPTION
    WHEN no_data_found THEN
        NULL;
    WHEN OTHERS THEN
        RAISE;
END;
/


CREATE OR REPLACE TRIGGER arc_fkarc_8_acd_checking BEFORE
    INSERT OR UPDATE OF a_uid, acct_type ON acd_checking
    FOR EACH ROW
DECLARE
    d VARCHAR2(2);
BEGIN
    SELECT
        a.acct_type
    INTO d
    FROM
        acd_safe_acnt a
    WHERE
        a.a_uid = :new.a_uid
        AND a.acct_type = :new.acct_type;

    IF ( d IS NULL OR d <> 'C' ) THEN

```

```

        raise_application_error(-20223, 'FK ACD_CHECKING_ACD_SAFE_ACNT_FK in Table
ACD_CHECKING violates Arc constraint on Table ACD_SAFE_ACNT - discriminator column
ACCT_TYPE doesn''t have value ''C'''
    );
END IF;

EXCEPTION
    WHEN no_data_found THEN
        NULL;
    WHEN OTHERS THEN
        RAISE;
END;
/

```

-- Oracle SQL Developer Data Modeler Summary Report:

-- CREATE TABLE	10
-- CREATE INDEX	2
-- ALTER TABLE	27
-- CREATE VIEW	0
-- ALTER VIEW	0
-- CREATE PACKAGE	0
-- CREATE PACKAGE BODY	0
-- CREATE PROCEDURE	0
-- CREATE FUNCTION	0
-- CREATE TRIGGER	6
-- ALTER TRIGGER	0
-- CREATE COLLECTION TYPE	0
-- CREATE STRUCTURED TYPE	0
-- CREATE STRUCTURED TYPE BODY	0
-- CREATE CLUSTER	0
-- CREATE CONTEXT	0
-- CREATE DATABASE	0
-- CREATE DIMENSION	0
-- CREATE DIRECTORY	0
-- CREATE DISK GROUP	0
-- CREATE ROLE	0
-- CREATE ROLLBACK SEGMENT	0
-- CREATE SEQUENCE	0
-- CREATE MATERIALIZED VIEW	0
-- CREATE MATERIALIZED VIEW LOG	0
-- CREATE SYNONYM	0
-- CREATE TABLESPACE	0
-- CREATE USER	0
--	
-- DROP TABLESPACE	0
-- DROP DATABASE	0
--	
-- REDACTION POLICY	0

```
--  
-- ORDS DROP SCHEMA          0  
-- ORDS ENABLE SCHEMA        0  
-- ORDS ENABLE OBJECT         0  
--  
-- ERRORS                     0  
-- WARNINGS                   0
```

5. DDL Code After Applying CHECK Constraint(s)

CHECK constraints for account type and loan type are already applied by generating DDL code from Oracle SQL Developer Data Modeler.

Now, a CHECK constraint is added to attribute ‘DEGREE_TYPE’ in ‘ACD_STUDENT’ being either it’s ‘Undergraduate’ or ‘Graduate’.

Below is the part of the code after applying this CHECK constraint.

```
ALTER TABLE acd_student
ADD CONSTRAINT chk_degree_type CHECK (degree_type IN ('Undergraduate',
'Graduate'));
```

Below is the entire DDL code with all the CHECK constraints.

```
-- Generated by Oracle SQL Developer Data Modeler 23.1.0.087.0806
-- at: 2024-03-28 18:37:14 EDT
-- site: Oracle Database 21c
-- type: Oracle Database 21c

-- predefined type, no DDL - MDSYS.SDO_Geometry

-- predefined type, no DDL - XMLTYPE

CREATE TABLE acd_checking (
    a_uid      NUMBER(10) NOT NULL,
    acct_type  VARCHAR2(2) NOT NULL,
    acct_no    NUMBER(10) NOT NULL,
    date_open   DATE NOT NULL,
    serv_charge NUMBER(6, 2) NOT NULL
);

COMMENT ON COLUMN acd_checking.a_uid IS
    'SAFE Unique Account ID';

COMMENT ON COLUMN acd_checking.acct_type IS
    'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_checking.acct_no IS
    'Checking Account Unique Account Number';

COMMENT ON COLUMN acd_checking.date_open IS
    'Account Opening Date';

COMMENT ON COLUMN acd_checking.serv_charge IS
    'Account Service Charge';

ALTER TABLE acd_checking ADD CONSTRAINT acd_checking_pk PRIMARY KEY ( a_uid,
```

```

        acct_type );
ALTER TABLE acd_checking ADD CONSTRAINT acd_checking_pkv1 UNIQUE ( acct_no );

CREATE TABLE acd_customer (
    c_id      NUMBER(5) NOT NULL,
    c_fname   VARCHAR2(30) NOT NULL,
    c_lname   VARCHAR2(30) NOT NULL,
    c_street  VARCHAR2(30) NOT NULL,
    c_city    VARCHAR2(30) NOT NULL,
    c_state   VARCHAR2(2) NOT NULL,
    c_zipcode VARCHAR2(5) NOT NULL,
    a_uid     NUMBER(10) NOT NULL,
    acct_type VARCHAR2(1) NOT NULL
);
COMMENT ON COLUMN acd_customer.c_id IS
    'Unique Customer ID';

COMMENT ON COLUMN acd_customer.c_fname IS
    'Customer First Name';

COMMENT ON COLUMN acd_customer.c_lname IS
    'Customer Last Name';

COMMENT ON COLUMN acd_customer.c_street IS
    'Customer Street - Address Line 1';

COMMENT ON COLUMN acd_customer.c_city IS
    'Customer City Location';

COMMENT ON COLUMN acd_customer.c_state IS
    'Customer State Code - Ex: New York: NY';

COMMENT ON COLUMN acd_customer.c_zipcode IS
    'Customer Zipcode';

CREATE UNIQUE INDEX acd_customer_idx ON
    acd_customer (
        a_uid
    ASC,
        acct_type
    ASC );
ALTER TABLE acd_customer ADD CONSTRAINT acd_customer_pk PRIMARY KEY ( c_id );

CREATE TABLE acd_home (
    a_uid      NUMBER(10) NOT NULL,
    acct_type  VARCHAR2(2) NOT NULL,
    hl_uid     NUMBER(5) NOT NULL,
    date_open  DATE NOT NULL,

```

```

        built_year NUMBER(4) NOT NULL
);

COMMENT ON COLUMN acd_home.a_uid IS
'SAFE Unique Account ID';

COMMENT ON COLUMN acd_home.acct_type IS
'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_home.hl_uid IS
'Home Loan Unique Loan ID';

COMMENT ON COLUMN acd_home.date_open IS
'Home Loan Account Opening Date';

COMMENT ON COLUMN acd_home.built_year IS
'Home Built Year';

ALTER TABLE acd_home ADD CONSTRAINT acd_home_pk PRIMARY KEY ( a_uid,
                                                               acct_type );

ALTER TABLE acd_home ADD CONSTRAINT acd_home_pkv1 UNIQUE ( hl_uid );

CREATE TABLE acd_institute (
    inst_code NUMBER(5) NOT NULL,
    inst_name VARCHAR2(30) NOT NULL
);

COMMENT ON COLUMN acd_institute.inst_code IS
'Unique Institute Code';

COMMENT ON COLUMN acd_institute.inst_name IS
'Institute Name (University Name)';

ALTER TABLE acd_institute ADD CONSTRAINT acd_institute_pk PRIMARY KEY ( inst_code
);

CREATE TABLE acd_insurance (
    ins_acct_no NUMBER(10) NOT NULL,
    ins_company VARCHAR2(30) NOT NULL,
    ins_street  VARCHAR2(30) NOT NULL,
    ins_city    VARCHAR2(30) NOT NULL,
    ins_state   VARCHAR2(2) NOT NULL,
    ins_zipcode VARCHAR2(5) NOT NULL,
    yearly_prm  NUMBER(8, 2) NOT NULL,
    hl_uid      NUMBER(5) NOT NULL
);

COMMENT ON COLUMN acd_insurance.ins_acct_no IS
'Home Insurance Unique Account Number';

```

```

COMMENT ON COLUMN acd_insurance.ins_company IS
    'Insurance Company Name';

COMMENT ON COLUMN acd_insurance.ins_street IS
    'Insurance Street Address';

COMMENT ON COLUMN acd_insurance.ins_city IS
    'Insurance City Location';

COMMENT ON COLUMN acd_insurance.ins_state IS
    'Insurance State Code- Ex: New York: NY';

COMMENT ON COLUMN acd_insurance.ins_zipcode IS
    'Insurance Company''s Zipcode';

COMMENT ON COLUMN acd_insurance.yearly_prm IS
    'Insurance Yearly Premium';

CREATE UNIQUE INDEX acd_insurance__idx ON
    acd_insurance (
        hl_uid
        ASC );
;

ALTER TABLE acd_insurance ADD CONSTRAINT acd_insurance_pk PRIMARY KEY ( ins_acct_no
);

CREATE TABLE acd_loan (
    a_uid      NUMBER(10) NOT NULL,
    acct_type  VARCHAR2(2) NOT NULL,
    acct_no    NUMBER(10) NOT NULL,
    loan_type  VARCHAR2(2) NOT NULL,
    loan_amt   NUMBER(10, 2) NOT NULL,
    loan_rate  NUMBER(4, 2) NOT NULL,
    loan_months NUMBER(3) NOT NULL,
    loan_payment NUMBER(8, 2) NOT NULL
);
;

ALTER TABLE acd_loan
    ADD CONSTRAINT ch_inh_acd_loan CHECK ( loan_type IN ( 'HL', 'PL', 'SL' ) );

COMMENT ON COLUMN acd_loan.a_uid IS
    'SAFE Unique Account ID';

COMMENT ON COLUMN acd_loan.acct_type IS
    'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_loan.acct_no IS
    'Loan Account Unique Account Number';

COMMENT ON COLUMN acd_loan.loan_type IS

```

```

'Loan Type – Student (SL)/ Home (HL)/ Personal (PL)';

COMMENT ON COLUMN acd_loan.loan_amt IS
    'OVERALL LOAN AMOUNT';

COMMENT ON COLUMN acd_loan.loan_rate IS
    'LOAN INTEREST RATE';

COMMENT ON COLUMN acd_loan.loan_months IS
    'LOAN DURATION';

COMMENT ON COLUMN acd_loan.loan_payment IS
    'LOAN REPAYMENT PER MONTH';

ALTER TABLE acd_loan ADD CONSTRAINT acd_loan_pk PRIMARY KEY ( a_uid,
    acct_type );

ALTER TABLE acd_loan ADD CONSTRAINT acd_loan_pkv1 UNIQUE ( acct_no,
    loan_type );

CREATE TABLE acd_personal (
    a_uid      NUMBER(10) NOT NULL,
    acct_type VARCHAR2(2) NOT NULL,
    pl_uid     NUMBER(5) NOT NULL,
    date_open  DATE NOT NULL
);

COMMENT ON COLUMN acd_personal.a_uid IS
    'SAFE Unique Account ID';

COMMENT ON COLUMN acd_personal.acct_type IS
    'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_personal.pl_uid IS
    'Personal Loan Unique ID';

COMMENT ON COLUMN acd_personal.date_open IS
    'Personal Loan Account Opening Date';

ALTER TABLE acd_personal ADD CONSTRAINT acd_personal_pk PRIMARY KEY ( a_uid,
    acct_type );

ALTER TABLE acd_personal ADD CONSTRAINT acd_personal_pkv1 UNIQUE ( pl_uid );

CREATE TABLE acd_safe_acnt (
    a_uid      NUMBER(10) NOT NULL,
    acct_type VARCHAR2(2) NOT NULL,
    acct_name VARCHAR2(30) NOT NULL,
    a_street   VARCHAR2(30) NOT NULL,
    a_city     VARCHAR2(30) NOT NULL,
    a_state    VARCHAR2(2) NOT NULL,

```

```

    a_zipcode VARCHAR2(5) NOT NULL
);

ALTER TABLE acd_safe_acnt
    ADD CONSTRAINT ch_inh_acd_safe_acnt CHECK ( acct_type IN ( 'C', 'HL', 'PL',
'S', 'SL','L' ) );

COMMENT ON COLUMN acd_safe_acnt.a_uid IS
    'SAFE Unique Account ID';

COMMENT ON COLUMN acd_safe_acnt.acct_type IS
    'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_safe_acnt.acct_name IS
    'Account Name';

COMMENT ON COLUMN acd_safe_acnt.a_street IS
    'Account Street Address';

COMMENT ON COLUMN acd_safe_acnt.a_city IS
    'Account City Location';

COMMENT ON COLUMN acd_safe_acnt.a_state IS
    'Account State Code – Ex: New York: NY';

COMMENT ON COLUMN acd_safe_acnt.a_zipcode IS
    'Account Zipcode';

ALTER TABLE acd_safe_acnt ADD CONSTRAINT acd_safe_acnt_pk PRIMARY KEY ( a_uid,
acct_type
);

CREATE TABLE acd_savings (
    a_uid      NUMBER(10) NOT NULL,
    acct_type  VARCHAR2(2) NOT NULL,
    acct_no    NUMBER(10) NOT NULL,
    date_open  DATE NOT NULL,
    intrst_rate NUMBER(4, 2) NOT NULL
);

COMMENT ON COLUMN acd_savings.a_uid IS
    'SAFE Unique Account ID';

COMMENT ON COLUMN acd_savings.acct_type IS
    'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_savings.acct_no IS
    'Savings Account Unique Account Number';

COMMENT ON COLUMN acd_savings.date_open IS

```

```

'Savings Account Opening Date';

COMMENT ON COLUMN acd_savings.intrst_rate IS
  'Interest Rate';

ALTER TABLE acd_savings ADD CONSTRAINT acd_savings_pk PRIMARY KEY ( a_uid,
  acct_type );

ALTER TABLE acd_savings ADD CONSTRAINT acd_savings_pkv1 UNIQUE ( acct_no );

CREATE TABLE acd_student (
  a_uid      NUMBER(10) NOT NULL,
  acct_type  VARCHAR2(2) NOT NULL,
  sl_uid     NUMBER NOT NULL,
  date_open   DATE NOT NULL,
  student_id VARCHAR2(6) NOT NULL,
  degree_type VARCHAR2(13) NOT NULL,
  grad_month  VARCHAR2(10) NOT NULL,
  grad_year   NUMBER(4) NOT NULL,
  inst_code   NUMBER(5) NOT NULL
);

COMMENT ON COLUMN acd_student.a_uid IS
  'SAFE Unique Account ID';

COMMENT ON COLUMN acd_student.acct_type IS
  'Account Type – Checking (C)/ Savings (S)/ Loan (L)';

COMMENT ON COLUMN acd_student.sl_uid IS
  'Student Loan Unique Loan ID';

COMMENT ON COLUMN acd_student.date_open IS
  'Student Loan Account Opening Date';

COMMENT ON COLUMN acd_student.student_id IS
  'Student ID From Institute';

COMMENT ON COLUMN acd_student.degree_type IS
  'Degree Type – Graduate / Undergraduate';

COMMENT ON COLUMN acd_student.grad_month IS
  'Graduation Month';

COMMENT ON COLUMN acd_student.grad_year IS
  'Graduation Year';

ALTER TABLE acd_student ADD CONSTRAINT acd_student_pk PRIMARY KEY ( a_uid,
  acct_type );

ALTER TABLE acd_student ADD CONSTRAINT acd_student_pkv1 UNIQUE ( sl_uid );

```

```

ALTER TABLE acd_student
ADD CONSTRAINT chk_degree_type CHECK (degree_type IN ('Undergraduate',
'Graduate'));

ALTER TABLE acd_checking
ADD CONSTRAINT acd_checking_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
                                                               acct_type )
REFERENCES acd_safe_acnt ( a_uid,
                           acct_type );

ALTER TABLE acd_customer
ADD CONSTRAINT acd_customer_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
                                                               acct_type )
REFERENCES acd_safe_acnt ( a_uid,
                           acct_type );

ALTER TABLE acd_home
ADD CONSTRAINT acd_home_acd_loan_fk FOREIGN KEY ( a_uid,
                                                               acct_type )
REFERENCES acd_loan ( a_uid,
                       acct_type );

ALTER TABLE acd_insurance
ADD CONSTRAINT acd_insurance_acd_home_fk FOREIGN KEY ( hl_uid )
REFERENCES acd_home ( hl_uid );

ALTER TABLE acd_loan
ADD CONSTRAINT acd_loan_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
                                                               acct_type )
REFERENCES acd_safe_acnt ( a_uid,
                           acct_type );

ALTER TABLE acd_personal
ADD CONSTRAINT acd_personal_acd_loan_fk FOREIGN KEY ( a_uid,
                                                               acct_type )
REFERENCES acd_loan ( a_uid,
                       acct_type );

ALTER TABLE acd_savings
ADD CONSTRAINT acd_savings_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
                                                               acct_type )
REFERENCES acd_safe_acnt ( a_uid,
                           acct_type );

ALTER TABLE acd_student
ADD CONSTRAINT acd_student_acd_institute_fk FOREIGN KEY ( inst_code )
REFERENCES acd_institute ( inst_code );

ALTER TABLE acd_student

```

```

ADD CONSTRAINT acd_student_acd_loan_fk FOREIGN KEY ( a_uid,
                                                    acct_type )
    REFERENCES acd_loan ( a_uid,
                           acct_type );

CREATE OR REPLACE TRIGGER arc_fkarc_7_acd_personal BEFORE
    INSERT OR UPDATE OF a_uid, acct_type ON acd_personal
    FOR EACH ROW
DECLARE
    d VARCHAR2(2);
BEGIN
    SELECT
        a.loan_type
    INTO d
    FROM
        acd_loan a
    WHERE
        a.a_uid = :new.a_uid
        AND a.acct_type = :new.acct_type;

    IF ( d IS NULL OR d <> 'PL' ) THEN
        raise_application_error(-20223, 'FK ACD_PERSONAL_ACD_LOAN_FK in Table
ACD_PERSONAL violates Arc constraint on Table ACD_LOAN - discriminator column
LOAN_TYPE doesn''t have value ''PL'''');
    END IF;

EXCEPTION
    WHEN no_data_found THEN
        NULL;
    WHEN OTHERS THEN
        RAISE;
END;
/

CREATE OR REPLACE TRIGGER arc_fkarc_7_acd_student BEFORE
    INSERT OR UPDATE OF a_uid, acct_type ON acd_student
    FOR EACH ROW
DECLARE
    d VARCHAR2(2);
BEGIN
    SELECT
        a.loan_type
    INTO d
    FROM
        acd_loan a
    WHERE
        a.a_uid = :new.a_uid
        AND a.acct_type = :new.acct_type;

    IF ( d IS NULL OR d <> 'SL' ) THEN

```

```

        raise_application_error(-20223, 'FK ACD_STUDENT_ACD_LOAN_FK in Table
ACD_STUDENT violates Arc constraint on Table ACD_LOAN - discriminator column
LOAN_TYPE doesn''t have value ''SL'''
    );
END IF;

EXCEPTION
  WHEN no_data_found THEN
    NULL;
  WHEN OTHERS THEN
    RAISE;
END;
/

CREATE OR REPLACE TRIGGER arc_fkarc_7_acd_home BEFORE
  INSERT OR UPDATE OF a_uid, acct_type ON acd_home
  FOR EACH ROW
DECLARE
  d VARCHAR2(2);
BEGIN
  SELECT
    a.loan_type
  INTO d
  FROM
    acd_loan a
  WHERE
    a.a_uid = :new.a_uid
    AND a.acct_type = :new.acct_type;

  IF ( d IS NULL OR d <> 'HL' ) THEN
    raise_application_error(-20223, 'FK ACD_HOME_ACD_LOAN_FK in Table ACD_HOME
violates Arc constraint on Table ACD_LOAN - discriminator column LOAN_TYPE doesn''t
have value ''HL'''
  );
END IF;

EXCEPTION
  WHEN no_data_found THEN
    NULL;
  WHEN OTHERS THEN
    RAISE;
END;
/

CREATE OR REPLACE TRIGGER arc_fkarc_8_acd_loan BEFORE
  INSERT OR UPDATE OF a_uid, acct_type ON acd_loan
  FOR EACH ROW
DECLARE
  d VARCHAR2(2);
BEGIN
  SELECT

```

```

        a.acct_type
INTO d
FROM
    acd_safe_acnt a
WHERE
    a.a_uid = :new.a_uid
    AND a.acct_type = :new.acct_type;

IF ( d IS NULL OR d <> 'L' ) THEN
    raise_application_error(-20223, 'FK ACD_LOAN_ACD_SAFE_ACNT_FK in Table
ACD_LOAN violates Arc constraint on Table ACD_SAFE_ACNT - discriminator column
ACCT_TYPE doesn''t have value ''L'''
);
END IF;

EXCEPTION
WHEN no_data_found THEN
    NULL;
WHEN OTHERS THEN
    RAISE;
END;
/

CREATE OR REPLACE TRIGGER arc_fkarc_8_acd_savings BEFORE
    INSERT OR UPDATE OF a_uid, acct_type ON acd_savings
    FOR EACH ROW
DECLARE
    d VARCHAR2(2);
BEGIN
    SELECT
        a.acct_type
    INTO d
    FROM
        acd_safe_acnt a
    WHERE
        a.a_uid = :new.a_uid
        AND a.acct_type = :new.acct_type;

    IF ( d IS NULL OR d <> 'S' ) THEN
        raise_application_error(-20223, 'FK ACD_SAVINGS_ACD_SAFE_ACNT_FK in Table
ACD_SAVINGS violates Arc constraint on Table ACD_SAFE_ACNT - discriminator column
ACCT_TYPE doesn''t have value ''S'''
);
    END IF;

EXCEPTION
WHEN no_data_found THEN
    NULL;
WHEN OTHERS THEN
    RAISE;
END;

```

```

/
CREATE OR REPLACE TRIGGER arc_fkarc_8_acd_checking BEFORE
  INSERT OR UPDATE OF a_uid, acct_type ON acd_checking
  FOR EACH ROW
DECLARE
  d VARCHAR2(2);
BEGIN
  SELECT
    a.acct_type
  INTO d
  FROM
    acd_safe_acnt a
  WHERE
    a.a_uid = :new.a_uid
    AND a.acct_type = :new.acct_type;

  IF ( d IS NULL OR d <> 'C' ) THEN
    raise_application_error(-20223, 'FK ACD_CHECKING_ACD_SAFE_ACNT_FK in Table
ACD_CHECKING violates Arc constraint on Table ACD_SAFE_ACNT - discriminator column
ACCT_TYPE doesn''t have value ''C'''');
  END IF;
EXCEPTION
  WHEN no_data_found THEN
    NULL;
  WHEN OTHERS THEN
    RAISE;
END;
/

```

-- Oracle SQL Developer Data Modeler Summary Report:

--	
-- CREATE TABLE	10
-- CREATE INDEX	2
-- ALTER TABLE	27
-- CREATE VIEW	0
-- ALTER VIEW	0
-- CREATE PACKAGE	0
-- CREATE PACKAGE BODY	0
-- CREATE PROCEDURE	0
-- CREATE FUNCTION	0
-- CREATE TRIGGER	6
-- ALTER TRIGGER	0
-- CREATE COLLECTION TYPE	0
-- CREATE STRUCTURED TYPE	0
-- CREATE STRUCTURED TYPE BODY	0
-- CREATE CLUSTER	0

-- CREATE CONTEXT	0
-- CREATE DATABASE	0
-- CREATE DIMENSION	0
-- CREATE DIRECTORY	0
-- CREATE DISK GROUP	0
-- CREATE ROLE	0
-- CREATE ROLLBACK SEGMENT	0
-- CREATE SEQUENCE	0
-- CREATE MATERIALIZED VIEW	0
-- CREATE MATERIALIZED VIEW LOG	0
-- CREATE SYNONYM	0
-- CREATE TABLESPACE	0
-- CREATE USER	0
--	
-- DROP TABLESPACE	0
-- DROP DATABASE	0
--	
-- REDACTION POLICY	0
--	
-- ORDS DROP SCHEMA	0
-- ORDS ENABLE SCHEMA	0
-- ORDS ENABLE OBJECT	0
--	
-- ERRORS	0
-- WARNINGS	0

6. Data Population DML Code

```
-- Begin transaction
START TRANSACTION;

-- If validation passes, commit transaction
-- COMMIT;
-- If validation fails, rollback transaction
-- ROLLBACK;
-- Inserting data into acd_institute table
INSERT INTO acd_institute (inst_code, inst_name)
VALUES
(1, 'University of Example'),
(2, 'Tech Institute of Innovation'),
(3, 'Global Science College'),
(4, 'Institute of Advanced Studies'),
(5, 'Metropolitan University'),
(6, 'National Arts Academy'),
(7, 'International Business School'),
(8, 'Center for Environmental Research'),
(9, 'Institute for Financial Studies'),
(10, 'Digital Media School'),
(11, 'School of Public Health'),
(12, 'Academy of Performing Arts'),
(13, 'Institute of Engineering and Technology'),
(14, 'School of International Relations'),
(15, 'College of Veterinary Medicine');

-- Inserting data into acd_safe_acnt table
INSERT INTO acd_safe_acnt (a_uid, acct_type, acct_name, a_street, a_city, a_state,
a_zipcode)
VALUES
(1, 'C', 'John Doe Checking', '123 Main St', 'Anytown', 'NY', '12345'),
(2, 'C', 'Jane Doe Checking', '456 Elm St', 'Anytown', 'NY', '12345'),
(3, 'C', 'Chris Smith Checking', '789 Pine St', 'Othertown', 'CA', '67890'),
(4, 'C', 'Pat Johnson Checking', '101 Maple Ave', 'Sometown', 'TX', '23456'),
(5, 'C', 'Alex Lee Checking', '202 Oak St', 'New City', 'FL', '34567'),
(6, 'C', 'Sammy O'Neil Checking', '303 Birch Rd', 'Old City', 'IL', '45678'),
(7, 'C', 'Morgan Casey Checking', '404 Cedar Ln', 'Capital City', 'VA', '56789'),
(8, 'C', 'Jamie Rivera Checking', '505 Pine Knoll Dr', 'Beach Town', 'NC',
'67890'),
(9, 'C', 'Casey Jordan Checking', '606 Spruce Ct', 'Lake City', 'MI', '78901'),
(10, 'C', 'Taylor Kim Checking', '707 Aspen Dr', 'Mountain Town', 'CO', '89012'),
(11, 'C', 'Jordan Pat Checking', '808 Redwood Blvd', 'River City', 'WA', '90123'),
(12, 'C', 'Sydney Morgan Checking', '909 Willow Way', 'Desert City', 'AZ',
'01234'),
(13, 'C', 'Riley Parker Checking', '1010 Maple Ln', 'Forest City', 'OR', '12345'),
(14, 'C', 'Casey Quinn Checking', '1111 Oak St', 'Island City', 'HI', '23456'),
(15, 'C', 'Alex Jordan Checking', '1212 Birch Ave', 'Cloud City', 'NM', '34567');
```

```

INSERT INTO acd_safe_acnt (a_uid, acct_type, acct_name, a_street, a_city, a_state,
a_zipcode)
VALUES
(1, 'S', 'John Doe Savings', '123 Main St', 'Anytown', 'NY', '12345'),
(2, 'S', 'Jane Doe Savings', '456 Oak St', 'Anytown', 'NY', '12346'),
(3, 'S', 'Chris Smith Savings', '789 Pine St', 'Midtown', 'CA', '92345'),
(4, 'S', 'Pat Johnson Savings', '101 Maple Ave', 'Uptown', 'TX', '52345'),
(5, 'S', 'Alex Lee Savings', '202 Oak Ln', 'Downtown', 'FL', '62345'),
(6, 'S', 'Sammy ONeil Savings', '303 Birch Rd', 'Eastside', 'IL', '72345'),
(7, 'S', 'Morgan Casey Savings', '404 Cedar Ln', 'Westside', 'VA', '82345'),
(8, 'S', 'Jamie Rivera Savings', '505 Pine Knoll Dr', 'Southtown', 'NC', '92345'),
(9, 'S', 'Casey Jordan Savings', '606 Spruce Ct', 'Northtown', 'MI', '02345'),
(10, 'S', 'Kim Lee Savings', '707 Elm St', 'Laketown', 'OH', '12395'),
(11, 'S', 'Alex Jordan Savings', '808 Willow Rd', 'Rivertown', 'WA', '12385'),
(12, 'S', 'Pat Kim Savings', '909 Maple Dr', 'Hilltown', 'CO', '12375'),
(13, 'S', 'Chris ONeil Savings', '1010 Oak Blvd', 'Beachside', 'NJ', '12365'),
(14, 'S', 'Jane Smith Savings', '1111 Pine Ave', 'Clifftown', 'ME', '12355'),
(15, 'S', 'John Lee Savings', '1212 Birch St', 'Greentown', 'PA', '12345');

```

```

INSERT INTO acd_safe_acnt (a_uid, acct_type, acct_name, a_street, a_city, a_state,
a_zipcode)
VALUES
(1, 'L', 'John Doe Loan', '123 Main St', 'Anytown', 'NY', '12345'),
(2, 'L', 'Jane Smith Loan', '234 Oak Ave', 'Springville', 'IL', '54321'),
(3, 'L', 'Chris Jones Loan', '345 Elm St', 'Laketown', 'CA', '67890'),
(4, 'L', 'Patricia Kim Loan', '456 Pine Rd', 'Mountville', 'TX', '23456'),
(5, 'L', 'Alex Lee Loan', '567 Spruce Ln', 'Hilltop', 'FL', '34567'),
(6, 'L', 'Sammy Oneil Loan', '678 Cedar Dr', 'Rivercity', 'IL', '45678'),
(7, 'L', 'Morgan Casey Loan', '789 Birch Way', 'Sunnyvale', 'VA', '56789'),
(8, 'L', 'Jamie Rivera Loan', '890 Maple Ave', 'Cloudtown', 'NC', '67890'),
(9, 'L', 'Casey Jordan Loan', '901 Oak St', 'Snowville', 'MI', '78901'),
(10, 'L', 'Kim Parker Loan', '123 Cherry Ln', 'Rainyville', 'OH', '89012'),
(11, 'L', 'Alex Thompson Loan', '234 Walnut St', 'Windyville', 'WA', '90123'),
(12, 'L', 'Pat Lee Loan', '345 Aspen Ct', 'Stormville', 'CO', '01234'),
(13, 'L', 'Chris Green Loan', '456 Redwood Cir', 'Sunnytown', 'NJ', '12349'),
(14, 'L', 'Jane Ford Loan', '567 Willow Rd', 'Cloudyville', 'ME', '23459'),
(15, 'L', 'John White Loan', '678 Palm Way', 'Starville', 'PA', '34569'),
(16, 'L', 'Emma Stone Loan', '789 Maple St', 'Moonville', 'KY', '45679'),
(17, 'L', 'Oliver Twist Loan', '890 Elm St', 'Sunsettown', 'NV', '56789'),
(18, 'L', 'Sophia Loren Loan', '901 Pine Ave', 'Dawntown', 'MT', '67899'),
(19, 'L', 'Liam Neeson Loan', '123 Oak Blvd', 'Twilighttown', 'ND', '78909'),
(20, 'L', 'Noah Webster Loan', '234 Birch Ln', 'Midnighttown', 'SD', '89019'),
(21, 'L', 'Ava Gardner Loan', '345 Cedar Dr', 'Noontown', 'WV', '90129'),
(22, 'L', 'Mia Farrow Loan', '456 Aspen Way', 'Dusktown', 'SC', '01239'),
(23, 'L', 'Isabella Rossellini Loan', '567 Redwood St', 'Dawntown', 'MS', '12359'),
(24, 'L', 'Ethan Hawke Loan', '678 Willow Rd', 'Noontown', 'MO', '23469'),
(25, 'L', 'Zoe Kazan Loan', '789 Palm Ln', 'Sunsettown', 'LA', '34579'),
(26, 'L', 'Lucas Hedges Loan', '890 Cherry St', 'Twilighttown', 'AR', '45689'),
(27, 'L', 'Chloe Sevigny Loan', '901 Walnut Ave', 'Midnighttown', 'AL', '56799'),
(28, 'L', 'Theo James Loan', '123 Spruce Blvd', 'Moonville', 'AK', '67809'),

```

```
(29, 'L', 'Shailene Woodley Loan', '234 Oak Ln', 'Sunsettown', 'VT', '78919'),
(30, 'L', 'Ansel Elgort Loan', '345 Maple Dr', 'Dawntown', 'ME', '89029'),
(31, 'L', 'Nat Wolff Loan', '456 Elm Way', 'Twilighttown', 'RI', '90139'),
(32, 'L', 'Laura Dern Loan', '567 Pine St', 'Midnighttown', 'NH', '01249'),
(33, 'L', 'Willem Dafoe Loan', '678 Birch Ave', 'Moonville', 'MT', '12369'),
(34, 'L', 'Nicole Kidman Loan', '789 Cedar Ln', 'Sunsettown', 'DE', '23479'),
(35, 'L', 'Keith Urban Loan', '890 Aspen St', 'Dawntown', 'ND', '34589'),
(36, 'L', 'Russell Crowe Loan', '901 Redwood Blvd', 'Noontown', 'SD', '45699'),
(37, 'L', 'Cate Blanchett Loan', '123 Willow Ln', 'Dusktown', 'WY', '56709'),
(38, 'L', 'Hugh Jackman Loan', '234 Palm Dr', 'Noontown', 'NM', '67819'),
(39, 'L', 'Patrick Stewart Loan', '345 Cherry Ave', 'Sunsettown', 'ID', '78929'),
(40, 'L', 'Ian McKellen Loan', '456 Walnut Way', 'Twilighttown', 'HI', '89039'),
(41, 'L', 'Benedict Cumberbatch Loan', '567 Spruce St', 'Midnighttown', 'CT',
'90149'),
(42, 'L', 'Martin Freeman Loan', '678 Oak Rd', 'Moonville', 'FL', '01259'),
(43, 'L', 'Orlando Bloom Loan', '789 Maple Ln', 'Sunsettown', 'AZ', '12379'),
(44, 'L', 'Elijah Wood Loan', '890 Elm St', 'Dawntown', 'TX', '23489'),
(45, 'L', 'Sean Astin Loan', '901 Pine Blvd', 'Twilighttown', 'CA', '34599);
```

-- Inserting data into acd_customer table

```
INSERT INTO acd_customer (c_id, c_fname, c_lname, c_street, c_city, c_state,
c_zipcode, a_uid, acct_type)
VALUES
(1, 'John', 'Doe', '123 Main St', 'Anytown', 'NY', '12345', 1, 'C'),
(2, 'Jane', 'Smith', '234 Oak Rd', 'Oldtown', 'TX', '23456', 2, 'C'),
(3, 'Chris', 'Brown', '345 Maple Ave', 'Laketown', 'CA', '34567', 3, 'C'),
(4, 'Patricia', 'Johnson', '456 Elm St', 'Hillside', 'FL', '45678', 4, 'C'),
(5, 'Michael', 'Davis', '567 Pine Rd', 'Riverview', 'IL', '56789', 5, 'C'),
(6, 'Linda', 'Martinez', '678 Spruce Ave', 'Mountain', 'VA', '67890', 6, 'C'),
(7, 'Robert', 'Wilson', '789 Birch St', 'Seaside', 'NC', '78901', 7, 'C'),
(8, 'Sarah', 'Moore', '890 Cedar Ln', 'Sunnyvale', 'MI', '89012', 8, 'C'),
(9, 'James', 'Taylor', '901 Willow Rd', 'Rainyville', 'OH', '90123', 9, 'C'),
(10, 'Jennifer', 'Anderson', '123 Oak St', 'Cloudtown', 'WA', '01234', 10, 'C'),
(11, 'William', 'Thomas', '234 Cherry Ln', 'Snowville', 'ME', '12349', 11, 'C'),
(12, 'Jessica', 'Jackson', '345 Walnut St', 'Stormville', 'CO', '23458', 12, 'C'),
(13, 'David', 'White', '456 Redwood Cir', 'Starville', 'PA', '34567', 13, 'C'),
(14, 'Emily', 'Harris', '567 Palm Way', 'Moonville', 'KY', '45676', 14, 'C'),
(15, 'Daniel', 'Lewis', '678 Cedar Path', 'Sunsettown', 'NV', '56785', 15, 'C');
```

INSERT INTO acd_checking (a_uid, acct_type, acct_no, date_open, serv_charge)

```
VALUES
(1, 'C', 123456789, '2024-03-28 12:00:00', 10.50),
(2, 'C', 223456789, '2024-03-28 12:00:00', 12.00),
(3, 'C', 323456789, '2024-03-28 12:00:00', 9.75),
(4, 'C', 423456789, '2024-03-28 12:00:00', 11.25),
(5, 'C', 523456789, '2024-03-28 12:00:00', 10.00),
(6, 'C', 623456789, '2024-03-28 12:00:00', 8.50),
```

```
(7, 'C', 723456789, '2024-03-28 12:00:00', 7.95),
(8, 'C', 823456789, '2024-03-28 12:00:00', 11.50),
(9, 'C', 923456789, '2024-03-28 12:00:00', 12.50),
(10, 'C', 1023456789, '2024-03-28 12:00:00', 10.75),
(11, 'C', 1123456789, '2024-03-28 12:00:00', 9.50),
(12, 'C', 1223456789, '2024-03-28 12:00:00', 11.00),
(13, 'C', 1323456789, '2024-03-28 12:00:00', 12.25),
(14, 'C', 1423456789, '2024-03-28 12:00:00', 10.25),
(15, 'C', 1523456789, '2024-03-28 12:00:00', 13.00);
```

-- Inserting data into acd_savings table

```
INSERT INTO acd_savings (a_uid, acct_type, acct_no, date_open, intrst_rate)
VALUES
(1, 'S', 987654321, '2024-03-28 12:00:00', 1.25),
(2, 'S', 987654322, '2024-03-28 12:00:00', 1.50),
(3, 'S', 987654323, '2024-03-28 12:00:00', 1.75),
(4, 'S', 987654324, '2024-03-28 12:00:00', 1.20),
(5, 'S', 987654325, '2024-03-28 12:00:00', 1.30),
(6, 'S', 987654326, '2024-03-28 12:00:00', 1.35),
(7, 'S', 987654327, '2024-03-28 12:00:00', 1.40),
(8, 'S', 987654328, '2024-03-28 12:00:00', 1.45),
(9, 'S', 987654329, '2024-03-28 12:00:00', 1.55),
(10, 'S', 987654330, '2024-03-28 12:00:00', 1.60),
(11, 'S', 987654331, '2024-03-28 12:00:00', 1.65),
(12, 'S', 987654332, '2024-03-28 12:00:00', 1.70),
(13, 'S', 987654333, '2024-03-28 12:00:00', 1.80),
(14, 'S', 987654334, '2024-03-28 12:00:00', 1.85),
(15, 'S', 987654335, '2024-03-28 12:00:00', 1.90);
```

```
INSERT INTO acd_loan (a_uid, acct_type, acct_no, loan_type, loan_amt, loan_rate,
loan_months, loan_payment)
```

```
VALUES
(1, 'L', 987554321, 'HL', 250000.00, 4.5, 360, 1263.37),
(2, 'L', 987554322, 'HL', 200000.00, 4.0, 360, 954.83),
(3, 'L', 987554323, 'HL', 300000.00, 4.75, 360, 1566.07),
(4, 'L', 987554324, 'HL', 350000.00, 4.25, 360, 1721.22),
(5, 'L', 987554325, 'HL', 150000.00, 3.75, 360, 694.67),
(6, 'L', 987554326, 'HL', 180000.00, 3.5, 360, 808.28),
(7, 'L', 987554327, 'HL', 220000.00, 4.6, 360, 1127.38),
(8, 'L', 987554328, 'HL', 260000.00, 4.8, 360, 1367.53),
(9, 'L', 987554329, 'HL', 280000.00, 4.2, 360, 1369.78),
(10, 'L', 987554330, 'HL', 320000.00, 3.9, 360, 1508.34),
(11, 'L', 987554331, 'HL', 240000.00, 4.4, 360, 1202.62),
(12, 'L', 987554332, 'HL', 260000.00, 5.0, 360, 1397.87),
(13, 'L', 987554333, 'HL', 210000.00, 3.8, 360, 976.22),
(14, 'L', 987554334, 'HL', 230000.00, 4.9, 360, 1219.22),
(15, 'L', 987554335, 'HL', 270000.00, 4.3, 360, 1337.44);
```

```

INSERT INTO acd_loan (a_uid, acct_type, acct_no, loan_type, loan_amt, loan_rate,
loan_months, loan_payment)
VALUES
(16, 'L', 987654332, 'PL', 50000.00, 5.5, 60, 955.65),
(17, 'L', 987654333, 'PL', 75000.00, 5.0, 120, 795.36),
(18, 'L', 987654334, 'PL', 100000.00, 4.75, 180, 769.04),
(19, 'L', 987654335, 'PL', 125000.00, 4.5, 240, 752.23),
(20, 'L', 987654336, 'PL', 150000.00, 4.25, 300, 740.19),
(21, 'L', 987654337, 'PL', 175000.00, 4.0, 360, 836.44),
(22, 'L', 987654338, 'PL', 200000.00, 3.75, 360, 926.23),
(23, 'L', 987654339, 'PL', 225000.00, 3.5, 360, 1010.35),
(24, 'L', 987654340, 'PL', 250000.00, 3.25, 360, 1089.53),
(25, 'L', 987654341, 'PL', 275000.00, 3.0, 360, 1163.94),
(26, 'L', 987654342, 'PL', 300000.00, 2.75, 360, 1224.63),
(27, 'L', 987654343, 'PL', 325000.00, 2.5, 360, 1281.63),
(28, 'L', 987654344, 'PL', 350000.00, 2.25, 360, 1335.00),
(29, 'L', 987654345, 'PL', 375000.00, 2.0, 360, 1385.79),
(30, 'L', 987654346, 'PL', 400000.00, 1.75, 360, 1433.98);

INSERT INTO acd_loan (a_uid, acct_type, acct_no, loan_type, loan_amt, loan_rate,
loan_months, loan_payment)
VALUES
(31, 'L', 987654348, 'SL', 25000.00, 4.25, 120, 253.34),
(32, 'L', 987654349, 'SL', 30000.00, 4.0, 180, 220.27),
(33, 'L', 987654350, 'SL', 35000.00, 3.75, 240, 202.02),
(34, 'L', 987654351, 'SL', 40000.00, 3.5, 300, 191.93),
(35, 'L', 987654352, 'SL', 45000.00, 3.25, 360, 196.77),
(36, 'L', 987654353, 'SL', 50000.00, 3.0, 360, 210.81),
(37, 'L', 987654354, 'SL', 55000.00, 2.75, 360, 224.05),
(38, 'L', 987654355, 'SL', 60000.00, 2.5, 360, 236.53),
(39, 'L', 987654356, 'SL', 65000.00, 2.25, 360, 248.26),
(40, 'L', 987654357, 'SL', 70000.00, 2.0, 360, 259.26),
(41, 'L', 987654358, 'SL', 75000.00, 1.75, 360, 269.54),
(42, 'L', 987654359, 'SL', 80000.00, 1.5, 360, 279.12),
(43, 'L', 987654360, 'SL', 85000.00, 1.25, 360, 288.01),
(44, 'L', 987654361, 'SL', 90000.00, 1.0, 360, 296.23),
(45, 'L', 987654362, 'SL', 95000.00, 0.75, 360, 303.78);

-- Inserting data into acd_student table
INSERT INTO acd_student (a_uid, acct_type, sl_uid, date_open, student_id,
degree_type, grad_month, inst_code, grad_year)
VALUES
(31, 'L', 654347, '2024-03-28 12:00:00', 'AB1236', 'Graduate', 'June', 3, 2025),
(32, 'L', 654348, '2024-03-28 12:00:00', 'AB1237', 'Undergraduate', 'May', 4,
2024),
(33, 'L', 654349, '2024-03-28 12:00:00', 'AB1238', 'Graduate', 'June', 1, 2025),
(34, 'L', 654350, '2024-03-28 12:00:00', 'AB1239', 'Undergraduate', 'May', 2,
2024),

```

```
(35, 'L', 654351, '2024-03-28 12:00:00', 'AB1240', 'Graduate', 'June', 3, 2025),
(36, 'L', 654352, '2024-03-28 12:00:00', 'AB1241', 'Undergraduate', 'May', 4, 2024),
(37, 'L', 654353, '2024-03-28 12:00:00', 'AB1242', 'Graduate', 'June', 1, 2025),
(38, 'L', 654354, '2024-03-28 12:00:00', 'AB1243', 'Undergraduate', 'May', 2, 2024),
(39, 'L', 654355, '2024-03-28 12:00:00', 'AB1244', 'Graduate', 'June', 3, 2025),
(40, 'L', 654356, '2024-03-28 12:00:00', 'AB1245', 'Undergraduate', 'May', 4, 2024),
(41, 'L', 654357, '2024-03-28 12:00:00', 'AB1246', 'Graduate', 'June', 1, 2025),
(42, 'L', 654358, '2024-03-28 12:00:00', 'AB1247', 'Undergraduate', 'May', 2, 2024),
(43, 'L', 654359, '2024-03-28 12:00:00', 'AB1248', 'Graduate', 'June', 3, 2025),
(44, 'L', 654360, '2024-03-28 12:00:00', 'AB1249', 'Undergraduate', 'May', 4, 2024),
(45, 'L', 654361, '2024-03-28 12:00:00', 'AB1250', 'Graduate', 'June', 1, 2025);
```

```
INSERT INTO acd_personal (a_uid, acct_type, pl_uid, date_open)
VALUES
(16, 'L', 123457, '2024-03-28 12:00:00'),
(17, 'L', 123458, '2024-03-28 12:00:00'),
(18, 'L', 123459, '2024-03-28 12:00:00'),
(19, 'L', 123460, '2024-03-28 12:00:00'),
(20, 'L', 123461, '2024-03-28 12:00:00'),
(21, 'L', 123462, '2024-03-28 12:00:00'),
(22, 'L', 123463, '2024-03-28 12:00:00'),
(23, 'L', 123464, '2024-03-28 12:00:00'),
(24, 'L', 123465, '2024-03-28 12:00:00'),
(25, 'L', 123466, '2024-03-28 12:00:00'),
(26, 'L', 123467, '2024-03-28 12:00:00'),
(27, 'L', 123468, '2024-03-28 12:00:00'),
(28, 'L', 123469, '2024-03-28 12:00:00'),
(29, 'L', 123470, '2024-03-28 12:00:00'),
(30, 'L', 123471, '2024-03-28 12:00:00');
```

```
INSERT INTO acd_home (a_uid, acct_type, hl_uid, date_open, built_year)
VALUES
(1, 'L', 987654321, '2024-03-28 12:00:00', 2005),
(2, 'L', 987654322, '2024-03-28 12:00:00', 1995),
(3, 'L', 987654323, '2024-03-28 12:00:00', 2010),
(4, 'L', 987654324, '2024-03-28 12:00:00', 2000),
(5, 'L', 987654325, '2024-03-28 12:00:00', 1980),
(6, 'L', 987654326, '2024-03-28 12:00:00', 1990),
(7, 'L', 987654327, '2024-03-28 12:00:00', 2015),
(8, 'L', 987654328, '2024-03-28 12:00:00', 2008),
(9, 'L', 987654329, '2024-03-28 12:00:00', 2003),
(10, 'L', 987654330, '2024-03-28 12:00:00', 1999),
(11, 'L', 987654331, '2024-03-28 12:00:00', 1985),
```

```

(12, 'L', 987654332, '2024-03-28 12:00:00', 1978),
(13, 'L', 987654333, '2024-03-28 12:00:00', 2012),
(14, 'L', 987654334, '2024-03-28 12:00:00', 1992),
(15, 'L', 987654335, '2024-03-28 12:00:00', 1988);

-- Inserting data into acd_insurance table
INSERT INTO acd_insurance (ins_acct_no, ins_company, ins_street, ins_city,
ins_state, ins_zipcode, yearly_prm, hl_uid)
VALUES
(123456789, 'Insurance Inc.', '456 Elm St', 'Anytown', 'NY', '12345', 1000.00,
987654321),
(123456790, 'SecureLife', '789 Pine St', 'Midtown', 'CA', '23456', 1050.00,
987654322),
(123456791, 'Guardian Co.', '101 Maple Ave', 'Uptown', 'TX', '34567', 950.00,
987654323),
(123456792, 'Safe Haven Ins.', '202 Oak Ln', 'Downtown', 'FL', '45678', 1100.00,
987654324),
(123456793, 'Trusty Assurance', '303 Birch Rd', 'Eastside', 'IL', '56789', 900.00,
987654325),
(123456794, 'Protector Plus', '404 Cedar Ln', 'Westside', 'VA', '67890', 1200.00,
987654326),
(123456795, 'Stability Shield', '505 Pine Knoll Dr', 'Southtown', 'NC', '78901',
1150.00, 987654327),
(123456796, 'Defender Ins.', '606 Spruce Ct', 'Lake City', 'MI', '89012', 1250.00,
987654328),
(123456797, 'SafetyNet Providers', '707 Elm St', 'Laketown', 'OH', '90123',
1300.00, 987654329),
(123456798, 'FutureSecure', '808 Willow Rd', 'Rivertown', 'WA', '01234', 950.00,
987654330),
(123456799, 'Harmony Insurances', '909 Maple Dr', 'Hilltown', 'CO', '12345',
1400.00, 987654331),
(123456800, 'PeaceMind Ins.', '111 Pine Ave', 'Clifftown', 'ME', '23456', 1350.00,
987654332),
(123456801, 'Unity Insurance Co.', '222 Cherry Ln', 'Rainyville', 'AZ', '34567',
1450.00, 987654333),
(123456802, 'Assured Tomorrow', '333 Oak St', 'Cloudtown', 'NM', '45678', 950.00,
987654334),
(123456803, 'Safeguard Associates', '444 Birch St', 'Greentown', 'UT', '56789',
1500.00, 987654335);

```

7. Count Of Records from Each Entity

ACD_CUSTOMER

```
1 •  select count(*) from acd_customer;
```

Result Grid	
count(*)	
▶	15

ACD_SAFE_ACNT

```
1 •  select count(*) from acd_safe_acnt;
```

Result Grid	
count(*)	
▶	75

ACD_SAVINGS

```
1 •  select count(*) from acd_savings;
```

Result Grid	
count(*)	
▶	15

ACD_CHECKING

A screenshot of the MySQL Workbench interface. The query editor at the top shows the command: `select count(*) from acd_checking;`. Below the editor is the results grid, which displays a single row with the value 15 under the column labeled "count(*)".

count(*)
15

ACD_LOAN

A screenshot of the MySQL Workbench interface. The query editor at the top shows the command: `select count(*) from acd_loan;`. Below the editor is the results grid, which displays a single row with the value 45 under the column labeled "count(*)".

count(*)
45

ACD_PERSONAL

A screenshot of the MySQL Workbench interface. The query editor at the top shows the command: `select count(*) from acd_personal;`. Below the editor is the results grid, which displays a single row with the value 15 under the column labeled "count(*)".

count(*)
15

ACD_HOME

A screenshot of the MySQL Workbench interface. The query editor at the top shows the command: `select count(*) from acd_home;`. Below the editor is the results grid, which displays one row of data: `count(*)` followed by the value `15`.

	count(*)
▶	15

ACD_STUDENT

A screenshot of the MySQL Workbench interface. The query editor at the top shows the command: `select count(*) from acd_student;`. Below the editor is the results grid, which displays one row of data: `count(*)` followed by the value `15`.

	count(*)
▶	15

ACD_INSTITUTE

A screenshot of the MySQL Workbench interface. The query editor at the top shows the command: `select count(*) from acd_institute;`. Below the editor is the results grid, which displays one row of data: `count(*)` followed by the value `15`.

	count(*)
▶	15

ACD_INSURANCE

The screenshot shows a MySQL Workbench interface. At the top, there's a toolbar with various icons. Below the toolbar, a query window displays the following SQL command:

```
1 •  select count(*) from acd_insurance;
```

Below the query window is a results grid. The grid has two columns: the first column is labeled "count(*)" and the second column contains the value "15". There are also navigation buttons for the results grid.

count(*)	15
▶	

8. Data Dictionary Queries Detailing All Tables

ACD_CUSTOMER

1 • SHOW FULL COLUMNS FROM acd_customer;

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
c_id	int(11)	NULL	NO	PRI	NULL		select,insert,update,references	Unique Customer ID
c_fname	varchar(30)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Customer First Name
c_lname	varchar(30)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Customer Last Name
c_street	varchar(30)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Customer Street - Address Line 1
c_city	varchar(30)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Customer City Location
c_state	varchar(2)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Customer State Code - Ex: New York: NY
c_zipcode	varchar(5)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Customer Zipcode
a_uid	bigint(20)	NULL	NO	MUL	NULL		select,insert,update,references	
acct_type	varchar(1)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	

ACD_SAFE_ACNT

1 • SHOW FULL COLUMNS FROM acd_safe_acnt;

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
a_uid	bigint(20)	NULL	NO	PRI	NULL		select,insert,update,references	SAFE Unique Account ID
acct_type	varchar(2)	utf8mb4_general_ci	NO	PRI	NULL		select,insert,update,references	Account Type - Checking (C)/ Savings (S)/ Loan...
acct_name	varchar(30)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Account Name
a_street	varchar(30)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Account Street Address
a_city	varchar(30)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Account City Location
a_state	varchar(2)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Account State Code - Ex: New York: NY
a_zipcode	varchar(5)	utf8mb4_general_ci	NO	NULL			select,insert,update,references	Account Zipcode

ACD_SAVINGS

1 • SHOW FULL COLUMNS FROM acd_savings;

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
a_uid	bigint(20)	NULL	NO	PRI	NULL		select,insert,update,references	SAFE Unique Account ID
acct_type	varchar(2)	utf8mb4_general_ci	NO	PRI	NULL		select,insert,update,references	Account Type - Checking (C)/ Savings (S)/ Loan...
acct_no	bigint(20)	NULL	NO	UNI	NULL		select,insert,update,references	Savings Account Unique Account Number
date_open	datetime	NULL	NO	NULL			select,insert,update,references	Savings Account Opening Date
intrst_rate	decimal(4,2)	NULL	NO	NULL			select,insert,update,references	Interest Rate

ACD_CHECKING

1 • SHOW FULL COLUMNS FROM acd_checking;

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
a_uid	bigint(20)	NULL	NO	PRI	NULL		select,insert,update,references	SAFE Unique Account ID
acct_type	varchar(2)	utf8mb4_general_ci	NO	PRI	NULL		select,insert,update,references	Account Type - Checking (C)/ Savings (S)/
acct_no	bigint(20)	NULL	NO	UNI	NULL		select,insert,update,references	Checking Account Unique Account Number
date_open	datetime	NULL	NO		NULL		select,insert,update,references	Account Opening Date
serv_charge	decimal(6,2)	NULL	NO		NULL		select,insert,update,references	Account Service Charge

ACD_LOAN

1 • SHOW FULL COLUMNS FROM acd_loan;

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
a_uid	bigint(20)	NULL	NO	PRI	NULL		select,insert,update,references	SAFE Unique Account ID
acct_type	varchar(2)	utf8mb4_general_ci	NO	PRI	NULL		select,insert,update,references	Account Type - Checking (C)/ Savings (S)/ Loan...
acct_no	bigint(20)	NULL	NO	MUL	NULL		select,insert,update,references	Loan Account Unique Account Number
loan_type	varchar(2)	utf8mb4_general_ci	NO		NULL		select,insert,update,references	Loan Type - Student (SL)/ Home (HL)/ Personal ...
loan_amt	decimal(10,2)	NULL	NO		NULL		select,insert,update,references	OVERALL LOAN AMOUNT
loan_rate	decimal(4,2)	NULL	NO		NULL		select,insert,update,references	LOAN INTEREST RATE
loan_months	smallint(6)	NULL	NO		NULL		select,insert,update,references	LOAN DURATION
loan_payment	decimal(8,2)	NULL	NO		NULL		select,insert,update,references	LOAN REPAYMENT PER MONTH

ACD_PERSONAL

1 • SHOW FULL COLUMNS FROM acd_personal;

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
a_uid	bigint(20)	NULL	NO	PRI	NULL		select,insert,update,references	SAFE Unique Account ID
acct_type	varchar(2)	utf8mb4_general_ci	NO	PRI	NULL		select,insert,update,references	Account Type - Checking (C)/ Savings (S)/ Loan...
pl_uid	int(11)	NULL	NO	UNI	NULL		select,insert,update,references	Personal Loan Unique ID
date_open	datetime	NULL	NO		NULL		select,insert,update,references	Personal Loan Account Opening Date

ACD_HOME

1 • SHOW FULL COLUMNS FROM acd_home;

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
a_uid	bigint(20)	NULL	NO	PRI	NULL		select,insert,update,references	SAFE Unique Account ID
acct_type	varchar(2)	utf8mb4_general_ci	NO	PRI	NULL		select,insert,update,references	Account Type - Checking (C) / Savings (S) / Loan...
hl_uid	int(11)	NULL	NO	UNI	NULL		select,insert,update,references	Home Loan Unique Loan ID
date_open	datetime	NULL	NO		NULL		select,insert,update,references	Home Loan Account Opening Date
built_year	smallint(6)	NULL	NO		NULL		select,insert,update,references	Home Built Year

ACD_STUDENT

1 • SHOW FULL COLUMNS FROM acd_student;

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
a_uid	bigint(20)	NULL	NO	PRI	NULL		select,insert,update,references	SAFE Unique Account ID
acct_type	varchar(2)	utf8mb4_general_ci	NO	PRI	NULL		select,insert,update,references	Account Type - Checking (C) / Savings (S) / Loan...
sl_uid	double	NULL	NO	UNI	NULL		select,insert,update,references	Student Loan Unique Loan ID
date_open	datetime	NULL	NO		NULL		select,insert,update,references	Student Loan Account Opening Date
student_id	varchar(6)	utf8mb4_general_ci	NO		NULL		select,insert,update,references	Student ID From Institute
degree_type	varchar(13)	utf8mb4_general_ci	NO		NULL		select,insert,update,references	Degree Type - Graduate / Undergraduate
grad_month	varchar(10)	utf8mb4_general_ci	NO		NULL		select,insert,update,references	Graduation Month
grad_year	smallint(6)	NULL	NO		NULL		select,insert,update,references	Graduation Year
inst_code	int(11)	NULL	NO	MUL	NULL		select,insert,update,references	

ACD_INSTITUTE

1 • SHOW FULL COLUMNS FROM acd_institute;

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
inst_code	int(11)	NULL	NO	PRI	NULL		select,insert,update,references	Unique Institute Code
inst_name	varchar(30)	utf8mb4_general_ci	NO		NULL		select,insert,update,references	Institute Name (University Name)

ACD_INSURANCE

1 • SHOW FULL COLUMNS FROM acd_insurance;

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
ins_acct_no	bigint(20)	NULL	NO	PRI	NULL		select,insert,update,references	Home Insurance Unique Account Number
ins_company	varchar(30)	utf8mb4_general_ci	NO		NULL		select,insert,update,references	Insurance Company Name
ins_street	varchar(30)	utf8mb4_general_ci	NO		NULL		select,insert,update,references	Insurance Street Address
ins_city	varchar(30)	utf8mb4_general_ci	NO		NULL		select,insert,update,references	Insurance City Location
ins_state	varchar(2)	utf8mb4_general_ci	NO		NULL		select,insert,update,references	Insurance State Code- Ex: New York: NY
ins_zipcode	varchar(5)	utf8mb4_general_ci	NO		NULL		select,insert,update,references	Insurance Company's Zipcode
yearly_prm	decimal(8,2)	NULL	NO		NULL		select,insert,update,references	Insurance Yearly Premium
hi_uid	int(11)	NULL	NO	UNI	NULL		select,insert,update,references	

9. MySQL Code – Database

```
-- SQLINES DEMO *** le SQL Developer Data Modeler 23.1.0.087.0806
-- SQLINES DEMO *** -03-28 18:37:14 EDT
-- SQLINES DEMO *** le Database 21c
-- SQLINES DEMO *** le Database 21c

-- SQLINES DEMO *** no DDL - MDSYS.SDO_GEOMETRY

-- SQLINES DEMO *** no DDL - XMLTYPE

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE acd_checking (
    a_uid      BIGINT NOT NULL COMMENT 'SAFE Unique Account ID',
    acct_type  VARCHAR(2) NOT NULL COMMENT 'Account Type - Checking (C)/ Savings
(S)/ Loan (L)',
    acct_no    BIGINT NOT NULL COMMENT 'Checking Account Unique Account Number',
    date_open   DATETIME NOT NULL COMMENT 'Account Opening Date',
    serv_charge DECIMAL(6, 2) NOT NULL COMMENT 'Account Service Charge'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_checking.a_uid IS
'SAFE Unique Account ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_checking.acct_type IS
'Account Type - Checking (C)/ Savings (S)/ Loan (L)'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_checking.acct_no IS
'Checking Account Unique Account Number'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_checking.date_open IS
'Account Opening Date'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_checking.serv_charge IS
'Account Service Charge'; */

ALTER TABLE acd_checking ADD CONSTRAINT acd_checking_pk PRIMARY KEY ( a_uid,
    acct_type );

ALTER TABLE acd_checking ADD CONSTRAINT acd_checking_pkv1 UNIQUE ( acct_no );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE acd_customer (
    c_id       INT NOT NULL COMMENT 'Unique Customer ID',
```

```

c_fname    VARCHAR(30) NOT NULL COMMENT 'Customer First Name',
c_lname    VARCHAR(30) NOT NULL COMMENT 'Customer Last Name',
c_street   VARCHAR(30) NOT NULL COMMENT 'Customer Street – Address Line 1',
c_city     VARCHAR(30) NOT NULL COMMENT 'Customer City Location',
c_state    VARCHAR(2) NOT NULL COMMENT 'Customer State Code – Ex: New York: NY',
c_zipcode  VARCHAR(5) NOT NULL COMMENT 'Customer Zipcode',
a_uid      BIGINT NOT NULL,
acct_type  VARCHAR(1) NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_customer.c_id IS
'Unique Customer ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_customer.c_fname IS
'Customer First Name'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_customer.c_lname IS
'Customer Last Name'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_customer.c_street IS
'Customer Street – Address Line 1'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_customer.c_city IS
'Customer City Location'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_customer.c_state IS
'Customer State Code – Ex: New York: NY'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_customer.c_zipcode IS
'Customer Zipcode'; */

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX acd_customer_idx ON
acd_customer (
    a_uid
ASC,
    acct_type
ASC );
ALTER TABLE acd_customer ADD CONSTRAINT acd_customer_pk PRIMARY KEY ( c_id );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE acd_home (
    a_uid      BIGINT NOT NULL COMMENT 'SAFE Unique Account ID',

```

```

    acct_type VARCHAR(2) NOT NULL COMMENT 'Account Type – Checking (C)/ Savings
(S)/ Loan (L)',  

    hl_uid      INT NOT NULL COMMENT 'Home Loan Unique Loan ID',  

    date_open   DATETIME NOT NULL COMMENT 'Home Loan Account Opening Date',  

    built_year  SMALLINT NOT NULL COMMENT 'Home Built Year'  

);  
  

/* Moved to CREATE TABLE  

COMMENT ON COLUMN acd_home.a_uid IS  

'SAFE Unique Account ID'; */  
  

/* Moved to CREATE TABLE  

COMMENT ON COLUMN acd_home.acct_type IS  

'Account Type – Checking (C)/ Savings (S)/ Loan (L)'; */  
  

/* Moved to CREATE TABLE  

COMMENT ON COLUMN acd_home.hl_uid IS  

'Home Loan Unique Loan ID'; */  
  

/* Moved to CREATE TABLE  

COMMENT ON COLUMN acd_home.date_open IS  

'Home Loan Account Opening Date'; */  
  

/* Moved to CREATE TABLE  

COMMENT ON COLUMN acd_home.built_year IS  

'Home Built Year'; */  
  

ALTER TABLE acd_home ADD CONSTRAINT acd_home_pk PRIMARY KEY ( a_uid,  

                                         acct_type );  
  

ALTER TABLE acd_home ADD CONSTRAINT acd_home_pkv1 UNIQUE ( hl_uid );  
  

-- SQLINES LICENSE FOR EVALUATION USE ONLY  

CREATE TABLE acd_institute (  

    inst_code INT NOT NULL COMMENT 'Unique Institute Code',  

    inst_name VARCHAR(30) NOT NULL COMMENT 'Institute Name (University Name)'  

);  
  

/* Moved to CREATE TABLE  

COMMENT ON COLUMN acd_institute.inst_code IS  

'Unique Institute Code'; */  
  

/* Moved to CREATE TABLE  

COMMENT ON COLUMN acd_institute.inst_name IS  

'Institute Name (University Name)'; */  
  

ALTER TABLE acd_institute ADD CONSTRAINT acd_institute_pk PRIMARY KEY ( inst_code
);  
  

-- SQLINES LICENSE FOR EVALUATION USE ONLY

```

```

CREATE TABLE acd_insurance (
    ins_acct_no BIGINT NOT NULL COMMENT 'Home Insurance Unique Account Number',
    ins_company VARCHAR(30) NOT NULL COMMENT 'Insurance Company Name',
    ins_street VARCHAR(30) NOT NULL COMMENT 'Insurance Street Address',
    ins_city    VARCHAR(30) NOT NULL COMMENT 'Insurance City Location',
    ins_state   VARCHAR(2) NOT NULL COMMENT 'Insurance State Code- Ex: New York:
NY',
    ins_zipcode VARCHAR(5) NOT NULL COMMENT 'Insurance Company''s Zipcode',
    yearly_prm  DECIMAL(8, 2) NOT NULL COMMENT 'Insurance Yearly Premium',
    hl_uid      INT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_insurance.ins_acct_no IS
'Home Insurance Unique Account Number'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_insurance.ins_company IS
'Insurance Company Name'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_insurance.ins_street IS
'Insurance Street Address'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_insurance.ins_city IS
'Insurance City Location'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_insurance.ins_state IS
'Insurance State Code- Ex: New York: NY'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_insurance.ins_zipcode IS
'Insurance Company''s Zipcode'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_insurance.yearly_prm IS
'Insurance Yearly Premium'; */

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE UNIQUE INDEX acd_insurance__idx ON
acd_insurance (
    hl_uid
    ASC );
;

ALTER TABLE acd_insurance ADD CONSTRAINT acd_insurance_pk PRIMARY KEY ( ins_acct_no
);

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE acd_loan (

```

```

    a_uid      BIGINT NOT NULL COMMENT 'SAFE Unique Account ID',
    acct_type  VARCHAR(2) NOT NULL COMMENT 'Account Type – Checking (C)/ Savings
(S)/ Loan (L)',
    acct_no    BIGINT NOT NULL COMMENT 'Loan Account Unique Account Number',
    loan_type   VARCHAR(2) NOT NULL COMMENT 'Loan Type – Student (SL)/ Home (HL)/
Personal (PL)',
    loan_amt   DECIMAL(10, 2) NOT NULL COMMENT 'OVERALL LOAN AMOUNT ',
    loan_rate   DECIMAL(4, 2) NOT NULL COMMENT 'LOAN INTEREST RATE',
    loan_months SMALLINT NOT NULL COMMENT 'LOAN DURATION',
    loan_payment DECIMAL(8, 2) NOT NULL COMMENT 'LOAN REPAYMENT PER MONTH'
);

ALTER TABLE acd_loan
ADD CONSTRAINT ch_inh_acd_loan CHECK ( loan_type IN ( 'HL', 'PL', 'SL' ) );

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_loan.a_uid IS
'SAFE Unique Account ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_loan.acct_type IS
'Account Type – Checking (C)/ Savings (S)/ Loan (L)'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_loan.acct_no IS
'Loan Account Unique Account Number'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_loan.loan_type IS
'Loan Type – Student (SL)/ Home (HL)/ Personal (PL)'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_loan.loan_amt IS
'OVERALL LOAN AMOUNT '; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_loan.loan_rate IS
'LOAN INTEREST RATE'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_loan.loan_months IS
'LOAN DURATION'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_loan.loan_payment IS
'LOAN REPAYMENT PER MONTH'; */

ALTER TABLE acd_loan ADD CONSTRAINT acd_loan_pk PRIMARY KEY ( a_uid,
                                                               acct_type );

ALTER TABLE acd_loan ADD CONSTRAINT acd_loan_pkv1 UNIQUE ( acct_no,
                                                               loan_type );

```

```

        loan_type );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE acd_personal (
    a_uid      BIGINT NOT NULL COMMENT 'SAFE Unique Account ID',
    acct_type VARCHAR(2) NOT NULL COMMENT 'Account Type – Checking (C)/ Savings
(S)/ Loan (L)',
    pl_uid      INT NOT NULL COMMENT 'Personal Loan Unique ID',
    date_open   DATETIME NOT NULL COMMENT 'Personal Loan Account Opening Date'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_personal.a_uid IS
'SAFE Unique Account ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_personal.acct_type IS
'Account Type – Checking (C)/ Savings (S)/ Loan (L)'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_personal.pl_uid IS
'Personal Loan Unique ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_personal.date_open IS
'Personal Loan Account Opening Date'; */

ALTER TABLE acd_personal ADD CONSTRAINT acd_personal_pk PRIMARY KEY ( a_uid,
                                                               acct_type );

ALTER TABLE acd_personal ADD CONSTRAINT acd_personal_pkv1 UNIQUE ( pl_uid );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE acd_safe_acnt (
    a_uid      BIGINT NOT NULL COMMENT 'SAFE Unique Account ID',
    acct_type VARCHAR(2) NOT NULL COMMENT 'Account Type – Checking (C)/ Savings
(S)/ Loan (L)',
    acct_name  VARCHAR(30) NOT NULL COMMENT 'Account Name',
    a_street   VARCHAR(30) NOT NULL COMMENT 'Account Street Address',
    a_city     VARCHAR(30) NOT NULL COMMENT 'Account City Location',
    a_state    VARCHAR(2) NOT NULL COMMENT 'Account State Code – Ex: New York: NY',
    a_zipcode  VARCHAR(5) NOT NULL COMMENT 'Account Zipcode'
);

ALTER TABLE acd_safe_acnt
    ADD CONSTRAINT ch_inh_acd_safe_acnt CHECK ( acct_type IN ( 'C', 'HL', 'PL',
'S', 'SL', 'L' ) );

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_safe_acnt.a_uid IS
'SAFE Unique Account ID'; */

```

```

'SAFE Unique Account ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_safe_acnt.acct_type IS
'Account Type – Checking (C)/ Savings (S)/ Loan (L)'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_safe_acnt.acct_name IS
'Account Name'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_safe_acnt.a_street IS
'Account Street Address'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_safe_acnt.a_city IS
'Account City Location'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_safe_acnt.a_state IS
'Account State Code – Ex: New York: NY'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_safe_acnt.a_zipcode IS
'Account Zipcode'; */

ALTER TABLE acd_safe_acnt ADD CONSTRAINT acd_safe_acnt_pk PRIMARY KEY ( a_uid,
                                                                acct_type
);

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE acd_savings (
    a_uid      BIGINT NOT NULL COMMENT 'SAFE Unique Account ID',
    acct_type  VARCHAR(2) NOT NULL COMMENT 'Account Type – Checking (C)/ Savings
(S)/ Loan (L)',
    acct_no    BIGINT NOT NULL COMMENT 'Savings Account Unique Account Number',
    date_open   DATETIME NOT NULL COMMENT 'Savings Account Opening Date',
    intrst_rate DECIMAL(4, 2) NOT NULL COMMENT 'Interest Rate'
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_savings.a_uid IS
'SAFE Unique Account ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_savings.acct_type IS
'Account Type – Checking (C)/ Savings (S)/ Loan (L)'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_savings.acct_no IS
'Savings Account Unique Account Number'; */

```

```

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_savings.date_open IS
'Savings Account Opening Date'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_savings.intrst_rate IS
'Interest Rate'; */

ALTER TABLE acd_savings ADD CONSTRAINT acd_savings_pk PRIMARY KEY ( a_uid,
acct_type );

ALTER TABLE acd_savings ADD CONSTRAINT acd_savings_pkv1 UNIQUE ( acct_no );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE acd_student (
    a_uid      BIGINT NOT NULL COMMENT 'SAFE Unique Account ID',
    acct_type  VARCHAR(2) NOT NULL COMMENT 'Account Type – Checking (C)/ Savings
(S)/ Loan (L)',
    sl_uid     DOUBLE NOT NULL COMMENT 'Student Loan Unique Loan ID',
    date_open   DATETIME NOT NULL COMMENT 'Student Loan Account Opening Date',
    student_id  VARCHAR(6) NOT NULL COMMENT 'Student ID From Institute',
    degree_type VARCHAR(13) NOT NULL COMMENT 'Degree Type – Graduate /
Undergraduate',
    grad_month  VARCHAR(10) NOT NULL COMMENT 'Graduation Month',
    grad_year    SMALLINT NOT NULL COMMENT 'Graduation Year',
    inst_code    INT NOT NULL
);

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_student.a_uid IS
'SAFE Unique Account ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_student.acct_type IS
'Account Type – Checking (C)/ Savings (S)/ Loan (L)'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_student.sl_uid IS
'Student Loan Unique Loan ID'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_student.date_open IS
'Student Loan Account Opening Date'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_student.student_id IS
'Student ID From Institute'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_student.degree_type IS

```

```

'Degree Type - Graduate / Undergraduate'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_student.grad_month IS
'Graduation Month'; */

/* Moved to CREATE TABLE
COMMENT ON COLUMN acd_student.grad_year IS
'Graduation Year'; */

ALTER TABLE acd_student ADD CONSTRAINT acd_student_pk PRIMARY KEY ( a_uid,
acct_type );

ALTER TABLE acd_student ADD CONSTRAINT acd_student_pkv1 UNIQUE ( sl_uid );

ALTER TABLE acd_student
ADD CONSTRAINT chk_degree_type CHECK (degree_type IN ('Undergraduate',
'Graduate'));

ALTER TABLE acd_checking
ADD CONSTRAINT acd_checking_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
acct_type )
REFERENCES acd_safe_acnt ( a_uid,
acct_type );

ALTER TABLE acd_customer
ADD CONSTRAINT acd_customer_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
acct_type )
REFERENCES acd_safe_acnt ( a_uid,
acct_type );

ALTER TABLE acd_home
ADD CONSTRAINT acd_home_acd_loan_fk FOREIGN KEY ( a_uid,
acct_type )
REFERENCES acd_loan ( a_uid,
acct_type );

ALTER TABLE acd_insurance
ADD CONSTRAINT acd_insurance_acd_home_fk FOREIGN KEY ( hl_uid )
REFERENCES acd_home ( hl_uid );

ALTER TABLE acd_loan
ADD CONSTRAINT acd_loan_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
acct_type )
REFERENCES acd_safe_acnt ( a_uid,
acct_type );

ALTER TABLE acd_personal
ADD CONSTRAINT acd_personal_acd_loan_fk FOREIGN KEY ( a_uid,
acct_type )
REFERENCES acd_loan ( a_uid,

```

```

        acct_type );

ALTER TABLE acd_savings
  ADD CONSTRAINT acd_savings_acd_safe_acnt_fk FOREIGN KEY ( a_uid,
                                                               acct_type )
    REFERENCES acd_safe_acnt ( a_uid,
                               acct_type );

ALTER TABLE acd_student
  ADD CONSTRAINT acd_student_acd_institute_fk FOREIGN KEY ( inst_code )
    REFERENCES acd_institute ( inst_code );

ALTER TABLE acd_student
  ADD CONSTRAINT acd_student_acd_loan_fk FOREIGN KEY ( a_uid,
                                                       acct_type )
    REFERENCES acd_loan ( a_uid,
                           acct_type );

-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_7_acd_personal;

DELIMITER //

CREATE TRIGGER arc_fkarc_7_acd_personal BEFORE INSERT ON acd_personal
FOR EACH ROW
BEGIN
  DECLARE d VARCHAR(2);
  DECLARE CONTINUE HANDLER FOR NOT FOUND BEGIN END;
  DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
    RESIGNAL;
  END;

  SELECT a.loan_type INTO d
  FROM acd_loan a
  WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

  IF (d IS NULL OR d <> 'PL') THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'FK ACD_PERSONAL_ACD_LOAN_FK in Table ACD_PERSONAL
violates Arc constraint on Table ACD_LOAN - discriminator column LOAN_TYPE doesn''t
have value ''PL'''';
    END IF;
END;
//

CREATE TRIGGER arc_fkarc_7_acd_personal_update BEFORE UPDATE ON acd_personal
FOR EACH ROW
BEGIN
  DECLARE d VARCHAR(2);
  DECLARE CONTINUE HANDLER FOR NOT FOUND BEGIN END;
  DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN

```

```

    RESIGNAL;
END;

SELECT a.loan_type INTO d
FROM acd_loan a
WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

IF (d IS NULL OR d <> 'PL') THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'FK ACD_PERSONAL_ACD_LOAN_FK in Table ACD_PERSONAL
violates Arc constraint on Table ACD_LOAN - discriminator column LOAN_TYPE doesn''t
have value ''PL'''';
END IF;
END;
//

DELIMITER ;

-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_7_acd_student;

DELIMITER //

CREATE TRIGGER arc_fkarc_7_acd_student BEFORE INSERT ON acd_student
FOR EACH ROW
BEGIN
    DECLARE d VARCHAR(2);
    DECLARE EXIT HANDLER FOR NOT FOUND BEGIN END;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        RESIGNAL;
    END;

    IF (NEW.a_uid IS NOT NULL AND NEW.acct_type IS NOT NULL) THEN
        SELECT a.loan_type INTO d
        FROM acd_loan a
        WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

        IF (d IS NULL OR d <> 'SL') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'FK ACD_STUDENT_ACD_LOAN_FK in Table ACD_STUDENT
violates Arc constraint on Table ACD_LOAN - discriminator column LOAN_TYPE doesn''t
have value ''SL'''';
        END IF;
    END IF;
END;
//

CREATE TRIGGER arc_fkarc_7_acd_student_update BEFORE UPDATE ON acd_student
FOR EACH ROW
BEGIN

```

```

DECLARE d VARCHAR(2);
DECLARE EXIT HANDLER FOR NOT FOUND BEGIN END;
DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
    RESIGNAL;
END;

IF (NEW.a_uid IS NOT NULL AND NEW.acct_type IS NOT NULL) THEN
    SELECT a.loan_type INTO d
    FROM acd_loan a
    WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

    IF (d IS NULL OR d <> 'SL') THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'FK ACD_STUDENT_ACD_LOAN_FK in Table ACD_STUDENT
violates Arc constraint on Table ACD_LOAN - discriminator column LOAN_TYPE doesn''t
have value ''SL'''';
    END IF;
END IF;
END;
//

DELIMITER ;

-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_7_acd_home;

DELIMITER //

CREATE TRIGGER arc_fkarc_7_acd_home BEFORE INSERT ON acd_home
FOR EACH ROW
BEGIN
    DECLARE d VARCHAR(2);
    DECLARE EXIT HANDLER FOR NOT FOUND BEGIN END;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        RESIGNAL;
    END;

    IF (NEW.a_uid IS NOT NULL AND NEW.acct_type IS NOT NULL) THEN
        SELECT a.loan_type INTO d
        FROM acd_loan a
        WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

        IF (d IS NULL OR d <> 'HL') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'FK ACD_HOME_ACD_LOAN_FK in Table ACD_HOME violates
Arc constraint on Table ACD_LOAN - discriminator column LOAN_TYPE doesn''t have
value ''HL'''';
        END IF;
    END IF;
END;

```

```

    END IF;
END;
//
CREATE TRIGGER arc_fkarc_7_acd_home_update BEFORE UPDATE ON acd_home
FOR EACH ROW
BEGIN
    DECLARE d VARCHAR(2);
    DECLARE EXIT HANDLER FOR NOT FOUND BEGIN END;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        RESIGNAL;
    END;

    IF (NEW.a_uid IS NOT NULL AND NEW.acct_type IS NOT NULL) THEN
        SELECT a.loan_type INTO d
        FROM acd_loan a
        WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

        IF (d IS NULL OR d <> 'HL') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'FK ACD_HOME_ACD_LOAN_FK in Table ACD_HOME violates
Arc constraint on Table ACD_LOAN – discriminator column LOAN_TYPE doesn''t have
value ''HL'''';
        END IF;
    END IF;
END;
//

DELIMITER ;

-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_8_acd_loan;

DELIMITER //

CREATE TRIGGER arc_fkarc_8_acd_loan BEFORE INSERT ON acd_loan
FOR EACH ROW
BEGIN
    DECLARE d VARCHAR(2);
    DECLARE EXIT HANDLER FOR NOT FOUND BEGIN END;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        RESIGNAL;
    END;

    IF (NEW.a_uid IS NOT NULL AND NEW.acct_type IS NOT NULL) THEN
        SELECT a.acct_type INTO d
        FROM acd_safe_acnt a
        WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

```

```

        IF (d IS NULL OR d <> 'L') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'FK ACD_LOAN_ACD_SAFE_ACNT_FK in Table ACD_LOAN
violates Arc constraint on Table ACD_SAFE_ACNT - discriminator column ACCT_TYPE
doesn''t have value ''L'''';
        END IF;
    END IF;
END;
//



CREATE TRIGGER arc_fkarc_8_acd_loan_update BEFORE UPDATE ON acd_loan
FOR EACH ROW
BEGIN
    DECLARE d VARCHAR(2);
    DECLARE EXIT HANDLER FOR NOT FOUND BEGIN END;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        RESIGNAL;
    END;

    IF (NEW.a_uid IS NOT NULL AND NEW.acct_type IS NOT NULL) THEN
        SELECT a.acct_type INTO d
        FROM acd_safe_acnt a
        WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

        IF (d IS NULL OR d <> 'L') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'FK ACD_LOAN_ACD_SAFE_ACNT_FK in Table ACD_LOAN
violates Arc constraint on Table ACD_SAFE_ACNT - discriminator column ACCT_TYPE
doesn''t have value ''L'''';
        END IF;
    END IF;
END;
//



DELIMITER ;




-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_8_acd_savings;

DELIMITER //



CREATE TRIGGER arc_fkarc_8_acd_savings BEFORE INSERT ON acd_savings
FOR EACH ROW
BEGIN
    DECLARE d VARCHAR(2);
    DECLARE EXIT HANDLER FOR NOT FOUND BEGIN END;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        RESIGNAL;
    END;

```

```

END;

IF (NEW.a_uid IS NOT NULL AND NEW.acct_type IS NOT NULL) THEN
    SELECT a.acct_type INTO d
    FROM acd_safe_acnt a
    WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

    IF (d IS NULL OR d <> 'S') THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'FK ACD_SAVINGS_ACD_SAFE_ACNT_FK in Table
ACD_SAVINGS violates Arc constraint on Table ACD_SAFE_ACNT – discriminator column
ACCT_TYPE doesn''t have value ''S'''';
    END IF;
END IF;
END;
//

CREATE TRIGGER arc_fkarc_8_acd_savings_update BEFORE UPDATE ON acd_savings
FOR EACH ROW
BEGIN
DECLARE d VARCHAR(2);
DECLARE EXIT HANDLER FOR NOT FOUND BEGIN END;
DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
    RESIGNAL;
END;

IF (NEW.a_uid IS NOT NULL AND NEW.acct_type IS NOT NULL) THEN
    SELECT a.acct_type INTO d
    FROM acd_safe_acnt a
    WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

    IF (d IS NULL OR d <> 'S') THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'FK ACD_SAVINGS_ACD_SAFE_ACNT_FK in Table
ACD_SAVINGS violates Arc constraint on Table ACD_SAFE_ACNT – discriminator column
ACCT_TYPE doesn''t have value ''S'''';
    END IF;
END IF;
END;
//

DELIMITER ;

-- SQLINES LICENSE FOR EVALUATION USE ONLY
DROP TRIGGER IF EXISTS arc_fkarc_8_acd_checking;

DELIMITER //

CREATE TRIGGER arc_fkarc_8_acd_checking BEFORE INSERT ON acd_checking

```

```

FOR EACH ROW
BEGIN
    DECLARE d VARCHAR(2);
    DECLARE EXIT HANDLER FOR NOT FOUND BEGIN END;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        RESIGNAL;
    END;

    IF (NEW.a_uid IS NOT NULL AND NEW.acct_type IS NOT NULL) THEN
        SELECT a.acct_type INTO d
        FROM acd_safe_acnt a
        WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

        IF (d IS NULL OR d <> 'C') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'FK ACD_CHECKING_ACD_SAFE_ACNT_FK in Table
ACD_CHECKING violates Arc constraint on Table ACD_SAFE_ACNT - discriminator column
ACCT_TYPE doesn''t have value ''C'''';
        END IF;
    END IF;
END;
//

CREATE TRIGGER arc_fkarc_8_acd_checking_update BEFORE UPDATE ON acd_checking
FOR EACH ROW
BEGIN
    DECLARE d VARCHAR(2);
    DECLARE EXIT HANDLER FOR NOT FOUND BEGIN END;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        RESIGNAL;
    END;

    IF (NEW.a_uid IS NOT NULL AND NEW.acct_type IS NOT NULL) THEN
        SELECT a.acct_type INTO d
        FROM acd_safe_acnt a
        WHERE a.a_uid = NEW.a_uid AND a.acct_type = NEW.acct_type;

        IF (d IS NULL OR d <> 'C') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'FK ACD_CHECKING_ACD_SAFE_ACNT_FK in Table
ACD_CHECKING violates Arc constraint on Table ACD_SAFE_ACNT - discriminator column
ACCT_TYPE doesn''t have value ''C'''';
        END IF;
    END IF;
END;
//

DELIMITER ;

```

-- SQLINES DEMO *** per Data Modeler Summary Report: