

GROUP : 1.9

ID : 201501019 : RAGHAV

201501041 : VIDISH

201501046 : YASH

DATABASE : COMPETITIVE_PROGRAMMING_PLATFORM

--DDL SCRIPT : RELATIONS

```
CREATE SCHEMA COMPETITIVE_PROGRAMMING_PLATFORM;  
SET SEARCH_PATH TO COMPETITIVE_PROGRAMMING_PLATFORM;
```

--RELATION_1: STATUS_TITLE

```
CREATE TABLE STATUS_TITLE  
(  
    TITLE VARCHAR(30),  
    START_RATING_RANGE INTEGER NOT NULL CHECK (START_RATING_RANGE >=  
0),  
    END_RATING_RANGE INTEGER NOT NULL CHECK (END_RATING_RANGE > 0),  
    CHECK (START_RATING_RANGE < END_RATING_RANGE),  
    PRIMARY KEY(TITLE)  
);
```

--RELATION_2: USERS

```
CREATE TABLE USERS  
(  
    HANDLE VARCHAR(20),  
    MAIL VARCHAR(50) NOT NULL UNIQUE,  
    FIRST_NAME VARCHAR(20) NOT NULL,  
    MIDDLE_NAME VARCHAR (20) NOT NULL,  
    LAST_NAME VARCHAR(20) NOT NULL,  
    BIRTH_DATE DATE NOT NULL,  
    GENDER CHAR(1) NOT NULL CHECK (GENDER = 'M' OR GENDER = 'F'),  
    REGISTER_DATE DATE NOT NULL,  
    ORGANIZATION VARCHAR(50) NULL,  
    CITY VARCHAR(20) NOT NULL,
```

```
COUNTRY VARCHAR(20) NOT NULL,  
RATING INTEGER NULL CHECK(RATING>=0 OR RATING IS NULL),  
PRIMARY KEY(HANDLE)  
);
```

--RELATION_3: **FOLLOWS**

```
CREATE TABLE FOLLOWS  
(  
    FOLLOWER VARCHAR(20),  
    FOLLOWING VARCHAR(20) CHECK(FOLLOWER<>FOLLOWING),  
    PRIMARY KEY(FOLLOWER , FOLLOWING),  
    FOREIGN KEY (FOLLOWER) REFERENCES USERS(HANDLE) ON DELETE  
CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (FOLLOWING) REFERENCES USERS(HANDLE) ON DELETE  
CASCADE ON UPDATE CASCADE  
);
```

--RELATION_4: **PROGRAMMES_IN**

```
CREATE TABLE PROGRAMMES_IN  
(  
    HANDLE VARCHAR(20),  
    LANGUAGE VARCHAR(20),  
    VERSION VARCHAR(10),  
    PRIMARY KEY(HANDLE,LANGUAGE,VERSION),  
    FOREIGN KEY (HANDLE) REFERENCES USERS(HANDLE) ON DELETE CASCADE  
ON UPDATE CASCADE  
);
```

--FUNCTION_1: **CHECK_VALID_DATETIME : RETURNS DIFFERENCE OF TWO DATE AND TIME OBJECTS.**

```
CREATE OR REPLACE FUNCTION CHECK_VALID_DATETIME(S_DATE DATE, S_TIME  
TIME(6) WITHOUT TIME ZONE, E_DATE DATE, E_TIME TIME(6) WITHOUT TIME ZONE)  
RETURNS INTEGER  
AS $BODY$  
    DECLARE
```

```

        DIFF INTEGER :=0;
BEGIN
    SELECT ((DATE_PART('day', E_DATE::timestamp - S_DATE::timestamp) * 24 +
    DATE_PART('hour', E_TIME::time - S_TIME::time)) * 60 +
    DATE_PART('minute', E_TIME::time - S_TIME::time)) INTO DIFF;

        RAISE NOTICE 'TIME DIFFERENCE = %',DIFF;
    RETURN DIFF;
END
$BODY$ LANGUAGE PLPGSQL;

```

--RELATION_5: CONTESTS

```

CREATE TABLE CONTESTS
(
    CONTEST_ID INTEGER CHECK (CONTEST_ID > 0),
    CONTEST_NAME VARCHAR(30) NOT NULL,
    START_DATE DATE NOT NULL,
    START_TIME TIME(6) WITHOUT TIME ZONE NOT NULL,
    END_DATE DATE NOT NULL,
    END_TIME TIME(6) WITHOUT TIME ZONE NOT NULL,
    PRIZE VARCHAR(20) NULL,
    CHECK(CHECK_VALID_DATETIME(START_DATE,START_TIME,END_DATE,END_TI
ME) >= 15),
    PRIMARY KEY(CONTEST_ID)
);

```

--RELATION_6: QUESTIONS

```

CREATE TABLE QUESTIONS
(
    QUESTION_ID CHAR(1) CHECK (QUESTION_ID>='A' AND QUESTION_ID<='Z'),
    CONTEST_ID INTEGER CHECK (CONTEST_ID > 0),
    DIFFICULTY_LEVEL VARCHAR(10) NOT NULL CHECK (DIFFICULTY_LEVEL IN
('LOW','MEDIUM','HARD','EXPERT')),
    MAXIMUM_SCORE INTEGER NOT NULL CHECK (MAXIMUM_SCORE>0),
    CREATOR_ID VARCHAR(20) NOT NULL,
    PRACTICE_SET_STATUS CHAR(1) NOT NULL
CHECK(PRACTICE_SET_STATUS='0' OR PRACTICE_SET_STATUS='1'),
    TEST_CASES INTEGER NOT NULL CHECK(TEST_CASES>0),

```

```

        CREATION_DATE DATE NOT NULL,
        CREATION_TIME TIME(6) WITHOUT TIME ZONE NOT NULL,
        PRIMARY KEY(CONTEST_ID,QUESTION_ID),
        FOREIGN KEY (CREATOR_ID) REFERENCES USERS(HANDLE) ON DELETE
        CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (CONTEST_ID) REFERENCES CONTESTS(CONTEST_ID) ON
        DELETE CASCADE ON UPDATE CASCADE
    );

```

--RELATION_7: HAS_CATEGORIES

```

CREATE TABLE HAS_CATEGORIES
(
    CONTEST_ID INTEGER CHECK(CONTEST_ID>0),
    QUESTION_ID CHAR(1) CHECK (QUESTION_ID>='A' AND QUESTION_ID<='Z'),
    CATEGORY VARCHAR(30),
    PRIMARY KEY(CONTEST_ID,QUESTION_ID,CATEGORY),
    FOREIGN KEY (CONTEST_ID,QUESTION_ID) REFERENCES
    QUESTIONS(CONTEST_ID,QUESTION_ID) ON DELETE CASCADE ON UPDATE CASCADE
);

```

--FUNCTION_2: CAN_PARTICIPATE : RETURNS TRUE IF HANDLE CAN PARTICIPATE ELSE FALSE.

```

CREATE OR REPLACE FUNCTION CAN_PARTICIPATE(CID INTEGER, R_DATE DATE,
R_TIME TIME(6) WITHOUT TIME ZONE ,HANDLE VARCHAR(20)) RETURNS BOOLEAN
AS $BODY$
    DECLARE
        QU QUESTIONS%ROWTYPE;
        CON CONTESTS%ROWTYPE;
        CS_DATE DATE;
        CS_TIME TIME(6) WITHOUT TIME ZONE;
        RES INTEGER;
    BEGIN
        FOR QU IN SELECT * FROM QUESTIONS WHERE CONTEST_ID=CID
        LOOP
            IF (QU.CREATOR_ID = HANDLE) THEN
                RETURN FALSE;
            END IF;
        END LOOP;
    END LOOP;

```

```

        SELECT * INTO CON FROM CONTESTS WHERE
CONTESTS.CONTEST_ID=CID;

        CS_DATE := CON.START_DATE;
        CS_TIME := CON.START_TIME;

        RAISE NOTICE 'DATE=% , TIME=%',CS_DATE , CS_TIME;

        SELECT
CHECK_VALID_DATETIME(R_DATE,R_TIME,CS_DATE,CS_TIME) INTO RES;
        RAISE NOTICE 'REGISTER RES = % MINUTES BEFORE CONTEST
STARTS.',RES;
        IF(RES>=0) THEN
            RETURN TRUE;
        ELSE
            RETURN FALSE;
        END IF;
    END
$BODY$ LANGUAGE PLPGSQL;

```

--RELATION_8: REGISTERS_IN

```

CREATE TABLE REGISTERS_IN
(
    HANDLE VARCHAR (20),
    CONTEST_ID INTEGER CHECK(CONTEST_ID>0),
    REGISTRATION_DATE DATE NOT NULL,
    REGISTRATION_TIME TIME(6) WITHOUT TIME ZONE NOT NULL,
    DISQUALIFIED_FLAG CHAR(1) NOT NULL CHECK(DISQUALIFIED_FLAG='0' OR
DISQUALIFIED_FLAG='1'),
    CONTEST_RATING INTEGER NULL CHECK(CONTEST_RATING>=0 OR
CONTEST_RATING IS NULL),
    PRIMARY KEY(HANDLE,CONTEST_ID),
    FOREIGN KEY (HANDLE) REFERENCES USERS(HANDLE) ON DELETE CASCADE
ON UPDATE CASCADE,
    FOREIGN KEY (CONTEST_ID) REFERENCES CONTESTS(CONTEST_ID) ON
DELETE CASCADE ON UPDATE CASCADE,
    CHECK(CAN_PARTICIPATE(CONTEST_ID,REGISTRATION_DATE,REGISTRATION_
TIME,HANDLE))
);

```

**--FUNCTION_3: CHECK_VALID_SUBMISSION_DATETIME : RETURNS TRUE IF
SUBMISSION DATE AND TIME IS GREATER THAN CONTEST START DATE AND TIME
ELSE FALSE**

```
CREATE OR REPLACE FUNCTION CHECK_VALID_SUBMISSION_DATETIME(S_DATE
DATE, S_TIME TIME(6) WITHOUT TIME ZONE, CID INTEGER) RETURNS BOOLEAN
AS $BODY$
    DECLARE
        C CONTESTS%ROWTYPE;
        CD DATE;
        CT TIME(6) WITHOUT TIME ZONE;
        RES INTEGER;
    BEGIN
        SELECT * INTO C FROM CONTESTS WHERE
CONTESTS.CONTEST_ID=CID;
        CD := C.START_DATE;
        CT := C.START_TIME;
        RAISE NOTICE 'C_DATE = % C_TIME = %' , CD , CT;
        SELECT CHECK_VALID_DATETIME(CD,CT,S_DATE,S_TIME) INTO RES;
        RAISE NOTICE 'SUB RES = % MINUTES AFTER CONTEST STARTS.' , RES;
        IF(RES>=0)THEN
            RETURN TRUE;
        ELSE
            RETURN FALSE;
        END IF;
    END
$BODY$ LANGUAGE PLPGSQL;
```

--RELATION_9: SUBMISSIONS

```
CREATE TABLE SUBMISSIONS
(
    SUBMISSION_ID INTEGER CHECK(SUBMISSION_ID>0),
    CONTEST_ID INTEGER NOT NULL CHECK(CONTEST_ID>0),
    QUESTION_ID CHAR(1) NOT NULL CHECK(QUESTION_ID>='A' AND
QUESTION_ID<='Z'),
    HANDLE VARCHAR(20) NOT NULL,
    LANGUAGE VARCHAR(20) NOT NULL,
    SUBMISSION_DATE DATE NOT NULL,
    SUBMISSION_TIME TIME(6) WITHOUT TIME ZONE NOT NULL,
```

```

        MEMORY INTEGER NOT NULL CHECK(MEMORY>=0),
        RUN_TIME REAL NOT NULL,
        PASS_CASES INTEGER NOT NULL CHECK(PASS_CASES>=0),
        SCORE DECIMAL(5,2) NOT NULL CHECK(SCORE>=0),
        STATUS VARCHAR (50) NOT NULL,
        CHECK(CHECK_VALID_SUBMISSION_DATETIME(SUBMISSION_DATE,SUBMISSIO
N_TIME,CONTEST_ID)),
        PRIMARY KEY(SUBMISSION_ID),
        FOREIGN KEY (CONTEST_ID,QUESTION_ID) REFERENCES
QUESTIONS(CONTEST_ID,QUESTION_ID) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (HANDLE) REFERENCES USERS(HANDLE) ON DELETE CASCADE
ON UPDATE CASCADE
);

```

--RELATION_10: LEADERBOARD

```

CREATE TABLE LEADERBOARD
(
    HANDLE VARCHAR(20),
    CONTEST_ID INTEGER CHECK(CONTEST_ID>0),
    SCORE DECIMAL(5,2) CHECK(SCORE>=0),
    STANDING INTEGER CHECK(STANDING>0),
    PRIMARY KEY(HANDLE,CONTEST_ID),
    FOREIGN KEY (HANDLE) REFERENCES USERS(HANDLE) ON DELETE CASCADE
ON UPDATE CASCADE,
    FOREIGN KEY (CONTEST_ID) REFERENCES CONTESTS(CONTEST_ID) ON
DELETE CASCADE ON UPDATE CASCADE
);

```

--RELATION_11: HAS_MEDAL

```

CREATE TABLE HAS_MEDALS
(
    HANDLE VARCHAR(20),
    CONTEST_ID INTEGER CHECK(CONTEST_ID>0),
    MEDAL_TYPE VARCHAR(10) CHECK(MEDAL_TYPE IN
('GOLD','SILVER','BRONZE')),
    PRIMARY KEY(HANDLE,CONTEST_ID),
    FOREIGN KEY (HANDLE) REFERENCES USERS(HANDLE) ON DELETE CASCADE
ON UPDATE CASCADE,

```

```
        FOREIGN KEY (CONTEST_ID) REFERENCES CONTESTS(CONTEST_ID) ON  
DELETE CASCADE ON UPDATE CASCADE  
);
```

--RELATION_12: EDITORIALS

```
CREATE TABLE EDITORIALS  
(  
    EDITORIAL_ID INTEGER CHECK(EDITORIAL_ID>0),  
    CONTEST_ID INTEGER CHECK(CONTEST_ID>0),  
    QUESTION_ID CHAR(1) CHECK(QUESTION_ID>='A' AND QUESTION_ID<='Z'),  
    CREATOR_ID VARCHAR(20) NOT NULL,  
    EXPLANATION VARCHAR(50) NOT NULL,  
    CODE VARCHAR (100) NULL,  
    PRIMARY KEY(EDITORIAL_ID,CONTEST_ID,QUESTION_ID),  
    FOREIGN KEY (CONTEST_ID,QUESTION_ID) REFERENCES  
QUESTIONS(CONTEST_ID,QUESTION_ID) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (CREATOR_ID) REFERENCES USERS(HANDLE) ON DELETE  
CASCADE ON UPDATE CASCADE  
);
```

--RELATION_13: DISCUSSIONS

```
CREATE TABLE DISCUSSIONS  
(  
    DISCUSSION_ID INTEGER CHECK(DISCUSSION_ID>0),  
    CONTEST_ID INTEGER CHECK(CONTEST_ID>0),  
    QUESTION_ID CHAR(1) CHECK(QUESTION_ID>='A' AND QUESTION_ID<='Z'),  
    HANDLE VARCHAR(20) NOT NULL,  
    DISCUSSION_DATE DATE NOT NULL,  
    DISCUSSION_TIME TIME(6) WITHOUT TIME ZONE NOT NULL,  
    COMMENT VARCHAR(50) NOT NULL,  
    PRIMARY KEY(DISCUSSION_ID,CONTEST_ID,QUESTION_ID),  
    FOREIGN KEY (CONTEST_ID,QUESTION_ID) REFERENCES  
QUESTIONS(CONTEST_ID,QUESTION_ID) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (HANDLE) REFERENCES USERS(HANDLE) ON DELETE CASCADE  
ON UPDATE CASCADE  
);
```