

## **Create Table Script**

We will use this script to create tables. We have included drop table statements to remove tables so that code can be rerun during the same session. The Drop table statements can be put in any order since we have used Cascade constraints to drop any referential integrity constraints referring to primary and unique keys in the dropped table.

```
DROP TABLE MEMBERSHIP CASCADE CONSTRAINTS;  
DROP TABLE CLASSES CASCADE CONSTRAINTS;  
DROP TABLE RESOURCEINFORMATION CASCADE CONSTRAINTS;  
DROP TABLE RESOURCES CASCADE CONSTRAINTS;  
DROP TABLE AUTHOR CASCADE CONSTRAINTS;  
DROP TABLE RESOURCEAUTHOR CASCADE CONSTRAINTS;  
DROP TABLE PUBLISHER CASCADE CONSTRAINTS;  
DROP TABLE RESOURCEPUBLISHER CASCADE CONSTRAINTS;  
DROP TABLE LOAN CASCADE CONSTRAINTS;  
DROP TABLE FINE CASCADE CONSTRAINTS;  
DROP TABLE EVALUATIONS_LOG;
```

```
ALTER SESSION SET NLS_DATE_FORMAT = 'dd-mm-yyyy';
```

```
CREATE TABLE MEMBERSHIP (  
    EMAIL VARCHAR2(55) CONSTRAINT PK_MEMBERSHIP PRIMARY KEY,  
    RCAP NUMBER(2) NOT NULL,  
    MNAME VARCHAR2(40),  
    ID NUMBER(10),  
    SD DATE NOT NULL,  
    FD DATE NOT NULL,  
    MTYPE VARCHAR2(7) NOT NULL,  
    SSTATUS VARCHAR2(1) NOT NULL,  
    CHECK (LENGTH(ID)=10),  
    CHECK (SD < FD),  
    CHECK ((MTYPE = 'Student' AND RCAP = 5) OR (MTYPE = 'Staff' AND RCAP =  
    10)) ,
```

```
CHECK (SSTATUS IN ('Y', 'N')),  
CHECK (EMAIL LIKE '%@%'),  
CHECK (EMAIL LIKE '%.%')  
);
```

```
CREATE TABLE CLASSES (  
CNUMBER NUMBER(4)  
CONSTRAINT PK_CLASS PRIMARY KEY,  
CNAME VARCHAR2(30) NOT NULL  
);
```

```
CREATE TABLE RESOURCEINFORMATION (  
ISBNEAN NUMBER(13) CONSTRAINT PK_RESOURCEINFORMATION  
PRIMARY KEY,  
CONSTRAINT CK_ISBNEAN  
CHECK ((LENGTH(ISBNEAN)>=10) AND (LENGTH(ISBNEAN)<=13)),  
CNUMBER NUMBER(4) NOT NULL,  
CONSTRAINT FK_RESOURCEINFORMATION_CNUMBER FOREIGN  
KEY(CNUMBER)  
REFERENCES CLASSES,  
RTYPE VARCHAR2(10) NOT NULL,  
TITLE VARCHAR2(100) NOT NULL,  
EDITION NUMBER(2),  
PYEAR NUMBER(4),  
LPERIOD NUMBER(2) NOT NULL,  
CONSTRAINT CK_RTYPE  
CHECK ((RTYPE IN ('CD', 'DVD', 'Book', 'Video', 'Journal')))  
);
```

```
CREATE TABLE RESOURCES (  
RID NUMBER(10)  
GENERATED BY DEFAULT ON NULL AS IDENTITY  
CONSTRAINT PK_RESOURCES PRIMARY KEY,  
ISBNEAN NUMBER(13) NOT NULL,  
CONSTRAINT FK_RESOURCES_ISBNEAN FOREIGN KEY(ISBNEAN)  
REFERENCES RESOURCEINFORMATION,  
LOCATION VARCHAR2(4) NOT NULL,
```

THCOPY NUMBER(2) NOT NULL  
);

CREATE TABLE AUTHOR (  
AID NUMBER(5)  
GENERATED BY DEFAULT ON NULL AS IDENTITY  
CONSTRAINT PK\_AUTHOR PRIMARY KEY,  
ANAME VARCHAR2(80) NOT NULL  
);

CREATE TABLE RESOURCEAUTHOR (  
ISBNEAN NUMBER(13),  
AID NUMBER(5),  
CONSTRAINT PK\_RESOURCEAUTHOR PRIMARY KEY(ISBNEAN,AID),  
CONSTRAINT FK\_RESOURCEAUTHOR\_ISBNEAN FOREIGN KEY(ISBNEAN)  
REFERENCES RESOURCEINFORMATION,  
CONSTRAINT FK\_RESOURCEAUTHOR\_AID FOREIGN KEY(AID)  
REFERENCES AUTHOR,  
CHECK ((LENGTH(ISBNEAN)>=10) AND (LENGTH(ISBNEAN)<=13))  
);

CREATE TABLE PUBLISHER (  
PID NUMBER(5)  
GENERATED BY DEFAULT ON NULL AS IDENTITY  
CONSTRAINT PK\_PUBLISHER PRIMARY KEY,  
PNAME VARCHAR2(80) NOT NULL  
);

CREATE TABLE RESOURCEPUBLISHER (  
ISBNEAN NUMBER(13),  
PID NUMBER(5),  
CONSTRAINT PK\_RESOURCEPUBLISHER PRIMARY KEY(ISBNEAN,PID),  
CONSTRAINT FK\_RESOURCEPUBLISHER\_ISBNEAN FOREIGN  
KEY(ISBNEAN)  
REFERENCES RESOURCEINFORMATION,  
CONSTRAINT FK\_RESOURCEPUBLISHER\_PID FOREIGN KEY(PID)  
REFERENCES PUBLISHER,  
CONSTRAINT CK\_ISBNEAN3

```
CHECK ((LENGTH(ISBNEAN)>=10) AND (LENGTH(ISBNEAN)<=13))
);
```

```
CREATE TABLE LOAN (
LREF NUMBER(10)
GENERATED BY DEFAULT ON NULL AS IDENTITY
CONSTRAINT PK_LOAN PRIMARY KEY,
EMAIL VARCHAR2(55),
CONSTRAINT FK_LOAN_LREF FOREIGN KEY(EMAIL)
REFERENCES MEMBERSHIP,
RID NUMBER(10),
CONSTRAINT FK_LOAN_RID FOREIGN KEY(RID)
REFERENCES RESOURCES,
RDATE DATE,
IDATE DATE NOT NULL,
CHECK (RDATE > IDATE)
);
```

```
CREATE TABLE FINE (
LREF NUMBER(10) CONSTRAINT PK_FINE PRIMARY KEY,
CONSTRAINT FK_FINE_LREF FOREIGN KEY(LREF)
REFERENCES LOAN,
DPAID DATE,
EMAIL VARCHAR2(55),
CONSTRAINT FK_FINE_EMAIL FOREIGN KEY(EMAIL)
REFERENCES MEMBERSHIP
);
```

```
CREATE TABLE EVALUATIONS_LOG (
log_date DATE,
action VARCHAR2(50)
);
```

### **Triggers**

Trigger to check if the resource you want to loan is available, if the borrower is suspended and if the borrower has reached the resource cap

```

CREATE OR REPLACE TRIGGER LOAN_START
  BEFORE INSERT
  ON LOAN
  FOR EACH ROW
DECLARE
  L_CNT NUMBER(2);
  L_CNT2 NUMBER(2);
BEGIN
  SELECT RESOURCECANBORROW
    INTO L_CNT2
    FROM (SELECT MEMBERSHIP.MNAME, MEMBERSHIP.EMAIL,
  (CASE
    WHEN SSTATUS = 'N' AND MEMBERSHIP.FD > SYSDATE THEN
  NVL(TABLE100.RESOURCECANBORROW1, MEMBERSHIP.RCAP)
    WHEN SSTATUS = 'Y' OR MEMBERSHIP.FD < SYSDATE THEN 0
    END) AS RESOURCECANBORROW
    FROM (SELECT LOAN.EMAIL,
    MEMBERSHIP.RCAP - COUNT(CASE WHEN RDATE IS NULL THEN 1 END)
  AS RESOURCECANBORROW1
    FROM LOAN
    INNER JOIN MEMBERSHIP
    ON LOAN.EMAIL = MEMBERSHIP.EMAIL
    GROUP BY LOAN.EMAIL, MEMBERSHIP.RCAP) TABLE100
    FULL OUTER JOIN MEMBERSHIP
    ON TABLE100.EMAIL = MEMBERSHIP.EMAIL) WHERE :NEW.EMAIL =
  EMAIL;
  SELECT COUNT(*)
    INTO L_CNT
    FROM (SELECT RID FROM (SELECT TABLE10.* FROM (SELECT
  RESOURCES.* , RESOURCEINFORMATION.TITLE FROM
  RESOURCEINFORMATION
  INNER JOIN RESOURCES
  ON RESOURCES.ISBNEAN = RESOURCEINFORMATION.ISBNEAN) TABLE10
  INNER JOIN
  (SELECT RID FROM RESOURCES
  MINUS
  SELECT RID FROM
  (SELECT RID FROM LOAN WHERE RDATE IS NULL
  )) TABLE20
  ON TABLE10.RID = TABLE20.RID) WHERE :NEW.RID = RID);

```

```

    IF L_CNT = 0 THEN
        RAISE_APPLICATION_ERROR(-20101, 'The resource has already been
loaned out');
    END IF;
    IF L_CNT2 = 0 THEN
        RAISE_APPLICATION_ERROR(-20102, 'The user can"t borrow anymore
resources');
    END IF;
END;

```

**Trigger to update the CNUMBER in RESOURCEINFORMATION when CNUMBER in CLASSES has been updated**

```

CREATE OR REPLACE TRIGGER CLASSES_CASCADE
BEFORE UPDATE OF CNUMBER
ON CLASSES
FOR EACH ROW
BEGIN
    UPDATE RESOURCEINFORMATION SET CNUMBER = :NEW.CNUMBER
    WHERE CNUMBER = :OLD.CNUMBER;
END;

```

**Trigger to update the AID in RESOURCEAUTHOR when AID in AUTHOR has been updated**

```

CREATE OR REPLACE TRIGGER RESOURCEAUTHOR_CASCADE
BEFORE UPDATE OF AID
ON AUTHOR
FOR EACH ROW
BEGIN
    UPDATE RESOURCEAUTHOR SET AID = :NEW.AID WHERE AID = :OLD.AID;
END;

```

**Trigger to update the PID in RESOURCEPUBLISHER when PID in PUBLISHER has been updated**

```

CREATE OR REPLACE TRIGGER RESOURCEPUBLISHER_CASCADE
BEFORE UPDATE OF PID
ON PUBLISHER
FOR EACH ROW
BEGIN

```

```
UPDATE RESOURCEPUBLISHER SET PID = :NEW.PID WHERE PID
= :OLD.PID;
END;
```

Creating a trigger to log table changes for the MEMBERSHIP table

### **EVAL\_CHANGE\_TRIGGER**

```
CREATE OR REPLACE TRIGGER EVAL_CHANGE_TRIGGER
AFTER INSERT OR UPDATE OR DELETE
ON MEMBERSHIP
DECLARE
log_action EVALUATIONS_LOG.action%TYPE;
BEGIN IF INSERTING THEN  log_action := 'Insert';
ELSIF UPDATING THEN  log_action := 'Update';
ELSIF DELETING THEN  log_action := 'Delete';
ELSE  DBMS_OUTPUT.PUT_LINE('This code is not reachable.');
```

END IF;

```
INSERT INTO EVALUATIONS_LOG (log_date, action)
VALUES (SYSDATE, log_action);
END;
```

### **Insert Data Script**

```
INSERT INTO MEMBERSHIP
VALUES('michelle.emmanuel@college.ac.uk',10,'Michelle
Emmanuel',1000000007, TO_DATE('01-09-2017', 'DD-MM-YYYY'),
TO_DATE('01-09-2024', 'DD-MM-YYYY'), 'Staff', 'N');
```

```
INSERT INTO MEMBERSHIP VALUES('louis.edwards@college.ac.uk',10,'Louis
Edwards',1000000320, TO_DATE('01-09-2017', 'DD-MM-YYYY'), TO_DATE('01-
09-2021', 'DD-MM-YYYY'), 'Staff', 'N');
```

```
INSERT INTO MEMBERSHIP VALUES('sh354@college.ac.uk',5,'Shamia
Hassan',2017433231, TO_DATE('01-09-2017', 'DD-MM-YYYY'), TO_DATE('30-
06-2021', 'DD-MM-YYYY'), 'Student', 'N');
```

```
INSERT INTO MEMBERSHIP VALUES('na021@college.ac.uk',5,'Nazia
Ali',2017445231, TO_DATE('01-09-2017', 'DD-MM-YYYY'), TO_DATE('30-06-
2021', 'DD-MM-YYYY'), 'Student', 'N');
```

```
INSERT INTO MEMBERSHIP VALUES('js354@college.ac.uk',5,'Joe Smith',
2019433231, TO_DATE('01-09-2019', 'DD-MM-YYYY'), TO_DATE('30-06-2022',
'DD-MM-YYYY'), 'Student', 'N');
```

```
INSERT INTO MEMBERSHIP VALUES('pf112@college.ac.uk',5,'Paula Fowler',
2019433233, TO_DATE('01-09-2019', 'DD-MM-YYYY'), TO_DATE('30-06-2022',
'DD-MM-YYYY'), 'Student', 'N');
```

```
INSERT INTO MEMBERSHIP VALUES('si.cheng@college.ac.uk',10,'Si
Cheng',1463919229, TO_DATE('01-09-2020', 'DD-MM-YYYY'), TO_DATE('01-09-
2022', 'DD-MM-YYYY'), 'Staff', 'N');
```

```
INSERT INTO MEMBERSHIP VALUES('mc081@college.ac.uk',5,'Matilda
Clarke',2020994332, TO_DATE('01-09-2020', 'DD-MM-YYYY'),TO_DATE('30-06-
2023', 'DD-MM-YYYY'), 'Student', 'N');
```

```
INSERT INTO CLASSES VALUES(0001,'Computer Science');
```

```
INSERT INTO CLASSES VALUES(0002,'History');
```

```
INSERT INTO CLASSES VALUES(0003,'Foreign Language Study');
```

```
INSERT INTO CLASSES VALUES(0004, 'Philosophy');
```

```
INSERT INTO CLASSES VALUES(0005,'Economy');
```

```
INSERT INTO CLASSES VALUES(0006,'Medical Science');
```

```
INSERT INTO RESOURCEINFORMATION
VALUES(9781292061184,0001,'Book','Database Systems',6,2010,14);
```

```
INSERT INTO RESOURCEINFORMATION
VALUES(9780198228589,0002,'Book','British history, 1815-1906',1,1991,14);
```

```
INSERT INTO RESOURCEINFORMATION
VALUES(9781907838101,0003,'CD','Chinese in steps. Student book 1',1,2011,2);
```

```
INSERT INTO RESOURCEINFORMATION
VALUES(9787560076195,0003,'Book','Little Oxford English-Chinese
Dictionary',9,2015,14);
```

```
INSERT INTO RESOURCEINFORMATION
VALUES(9780415337977,0004,'Book','Philosophy : basic readings',2,2005,14);
```



```
INSERT INTO RESOURCEINFORMATION  
VALUES(9770013061237,0005,'Journal','The Economist Issue NOVEMBER  
2020',1,2020,0);
```

```
INSERT INTO RESOURCEINFORMATION  
VALUES(9780472001095,0006,'Book','Virus',1,1959,14);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9781292061184, '0012', 1);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9781292061184, '0012', 2);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9781292061184, '0012', 3);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9780198228589, '1120', 1);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9781907838101, '0013', 1);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9787560076195, '0017', 1);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9780415337977, '0095', 1);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9770013061237, '0031', 1);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9780472001095, '0032', 1);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9780472001095, '0032', 2);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9780472001095, '0032', 3);
```

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES  
(9780472001095, '0032', 4);
```

```
INSERT INTO AUTHOR(ANAME) VALUES ('Thomas Connolly');
```

```
INSERT INTO AUTHOR (ANAME) VALUES ('Carolyn Begg');
```

```
INSERT INTO AUTHOR (ANAME) VALUES ('Norman McCord');
```

```
INSERT INTO AUTHOR (ANAME) VALUES ('George Zhang');
```

```
INSERT INTO AUTHOR (ANAME) VALUES ('Linda Li');
```

```
INSERT INTO AUTHOR (ANAME) VALUES ('Lik Suen');
```

INSERT INTO AUTHOR (ANAME) VALUES ('Oxford University Press');  
INSERT INTO AUTHOR (ANAME) VALUES ('Nigel Warburton');  
INSERT INTO AUTHOR (ANAME) VALUES ('The Economist');  
INSERT INTO AUTHOR (ANAME) VALUES ('Wolfhard Weidel');

INSERT INTO RESOURCEAUTHOR VALUES (9781292061184, 1);  
INSERT INTO RESOURCEAUTHOR VALUES (9781292061184, 2);  
INSERT INTO RESOURCEAUTHOR VALUES (9780198228589, 3);  
INSERT INTO RESOURCEAUTHOR VALUES (9781907838101, 4);  
INSERT INTO RESOURCEAUTHOR VALUES (9781907838101, 5);  
INSERT INTO RESOURCEAUTHOR VALUES (9781907838101, 6);  
INSERT INTO RESOURCEAUTHOR VALUES (9787560076195, 7);  
INSERT INTO RESOURCEAUTHOR VALUES (9780415337977, 8);  
INSERT INTO RESOURCEAUTHOR VALUES (9770013061237, 9);  
INSERT INTO RESOURCEAUTHOR VALUES (9780472001095, 10);

INSERT INTO PUBLISHER (PNAME) VALUES ('Pearson');  
INSERT INTO PUBLISHER (PNAME) VALUES ('Oxford University Press');  
INSERT INTO PUBLISHER (PNAME) VALUES ('Sinolingua London Ltd.');

INSERT INTO PUBLISHER (PNAME) VALUES ('Foreign Language Teaching and Research Press');

INSERT INTO PUBLISHER (PNAME) VALUES ('Routledge');

INSERT INTO PUBLISHER (PNAME) VALUES ('The Economist Group');

INSERT INTO PUBLISHER (PNAME) VALUES ('University of Michigan Press');

INSERT INTO RESOURCEPUBLISHER VALUES (9781292061184, 1);  
INSERT INTO RESOURCEPUBLISHER VALUES (9780198228589, 2);  
INSERT INTO RESOURCEPUBLISHER VALUES (9781907838101, 3);  
INSERT INTO RESOURCEPUBLISHER VALUES (9787560076195, 2);  
INSERT INTO RESOURCEPUBLISHER VALUES (9787560076195, 4);  
INSERT INTO RESOURCEPUBLISHER VALUES (9780415337977, 5);  
INSERT INTO RESOURCEPUBLISHER VALUES (9770013061237, 6);  
INSERT INTO RESOURCEPUBLISHER VALUES (9780472001095, 7);

INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES  
('michelle.emmanuel@college.ac.uk', 1, TO\_DATE('01-09-2020', 'DD-MM-YYYY'),  
TO\_DATE('17-09-2020', 'DD-MM-YYYY'));

```

INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('michelle.emmanuel@college.ac.uk', 10, TO_DATE('01-09-2020', 'DD-MM-YYYY'), TO_DATE('03-09-2020', 'DD-MM-YYYY'));
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('sh354@college.ac.uk', 2, TO_DATE('03-09-2020', 'DD-MM-YYYY'), TO_DATE('05-09-2020', 'DD-MM-YYYY'));
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('na021@college.ac.uk', 5, TO_DATE('04-09-2020', 'DD-MM-YYYY'), TO_DATE('09-09-2020', 'DD-MM-YYYY'));
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('mc081@college.ac.uk', 9, TO_DATE('05-09-2020', 'DD-MM-YYYY'), TO_DATE('09-09-2020', 'DD-MM-YYYY'));
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('mc081@college.ac.uk', 5, TO_DATE('10-09-2020', 'DD-MM-YYYY'), TO_DATE('12-09-2020', 'DD-MM-YYYY'));
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('louis.edwards@college.ac.uk', 11, TO_DATE('07-11-2020', 'DD-MM-YYYY'), NULL);
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('louis.edwards@college.ac.uk', 1, TO_DATE('07-11-2020', 'DD-MM-YYYY'), TO_DATE('20-11-2020', 'DD-MM-YYYY'));
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('louis.edwards@college.ac.uk', 6, TO_DATE('07-11-2020', 'DD-MM-YYYY'), TO_DATE('20-11-2020', 'DD-MM-YYYY'));
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('louis.edwards@college.ac.uk', 7, TO_DATE('07-11-2020', 'DD-MM-YYYY'), NULL);
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('na021@college.ac.uk', 9, TO_DATE('10-11-2020', 'DD-MM-YYYY'), TO_DATE('30-11-2020', 'DD-MM-YYYY'));
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('na021@college.ac.uk', 5, TO_DATE('15-11-2020', 'DD-MM-YYYY'), TO_DATE('17-11-2020', 'DD-MM-YYYY'));
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('sh354@college.ac.uk', 3, TO_DATE('16-11-2020', 'DD-MM-YYYY'), TO_DATE('20-11-2020', 'DD-MM-YYYY'));
INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES
('sh354@college.ac.uk', 5, TO_DATE('01-12-2020', 'DD-MM-YYYY'), NULL);

INSERT INTO FINE VALUES (1 ,TO_DATE('17-09-2020', 'DD-MM-YYYY'), 'michelle.emmanuel@college.ac.uk');

```

```

INSERT INTO FINE VALUES (4, TO_DATE('09-09-2020', 'DD-MM-YYYY'),
'na021@college.ac.uk');
INSERT INTO FINE VALUES (7, NULL, 'louis.edwards@college.ac.uk');
INSERT INTO FINE VALUES (10, NULL, 'louis.edwards@college.ac.uk');
INSERT INTO FINE VALUES (11, TO_DATE('02-12-2020', 'DD-MM-YYYY'),
'na021@college.ac.uk');
INSERT INTO FINE VALUES (14, NULL, 'sh354@college.ac.uk');

```

```

UPDATE MEMBERSHIP SET SSTATUS = 'Y' WHERE EMAIL =
'louis.edwards@college.ac.uk';
UPDATE MEMBERSHIP SET SSTATUS = 'Y' WHERE EMAIL =
'sh354@college.ac.uk';

```

## **Views**

Popularity list views for 09/2020:

```

CREATE VIEW POPULARLIST AS
(SELECT TITLE, COUNT(TITLE) AS TIMESBORROWED FROM
(SELECT RESOURCES.ISBNEAN, LOAN.*, RESOURCEINFORMATION.TITLE
FROM RESOURCES
INNER JOIN LOAN
ON RESOURCES.RID = LOAN.RID
INNER JOIN RESOURCEINFORMATION
ON RESOURCES.ISBNEAN = RESOURCEINFORMATION.ISBNEAN
)
WHERE IDATE BETWEEN TO_DATE('01-09-2020','dd-mm-yyyy') AND
TO_DATE('30-09-2020','dd-mm-yyyy')
GROUP BY TITLE
);

```

BOOKS AVAILABLE VIEW:

```

CREATE VIEW COPIESAVAILABLE AS
(SELECT TABLE1.* FROM (SELECT RESOURCES.* ,
RESOURCEINFORMATION.TITLE FROM RESOURCEINFORMATION
INNER JOIN RESOURCES
ON RESOURCES.ISBNEAN = RESOURCEINFORMATION.ISBNEAN) TABLE1

```

```

INNER JOIN
(SELECT RID FROM RESOURCES
MINUS
SELECT RID FROM
(SELECT RID FROM LOAN WHERE RDATE IS NULL
)) TABLE2
ON TABLE1.RID = TABLE2.RID
);

```

Fine amount

```

CREATE OR REPLACE VIEW FINESTATUS AS
(SELECT FINEINFO.*, FINE.DPAID FROM
(SELECT EMAIL,LREF, IDATE, IDATE+LPERIOD AS DUEDATE,
RDATEorTODAY, RDATEorTODAY-(IDATE+LPERIOD) AS FINEAMOUNT
FROM (SELECT LOAN.LREF, LOAN.EMAIL, LOAN.IDATE,
TO_DATE(NVL(LOAN.RDATE,SYSDATE),' dd-mm-yyyy ') AS RDATEorTODAY,
RES1.*
FROM LOAN
INNER JOIN (SELECT RESOURCES.RID, RESOURCEINFORMATION.* FROM
RESOURCES INNER JOIN RESOURCEINFORMATION ON
RESOURCES.ISBNEAN = RESOURCEINFORMATION.ISBNEAN) RES1 ON
LOAN.RID = RES1.RID)
WHERE RDATEorTODAY-(IDATE+LPERIOD) >0) FINEINFO
INNER JOIN FINE
ON FINEINFO.LREF = FINE.LREF)

```

TOTAL FINE NOT PAID:

```

CREATE OR REPLACE VIEW TOTALFINENOTPAID AS
(SELECT EMAIL, SUM(FINEAMOUNT) AS TOTALFINE,
CASE
WHEN SUM(FINEAMOUNT) >= 10 THEN 'Y'
WHEN SUM(FINEAMOUNT) <10 THEN 'N'
END AS SSTATUS
FROM FINESTATUS
WHERE DPAID IS NULL
GROUP BY EMAIL)

```

## **Queries**

-- Find a Member with a particular name--

```
SELECT *  
FROM MEMBERSHIP  
WHERE (MNAME LIKE '%Edwards');
```

-- Calculate the average length of a loan(DAYS)--

```
SELECT ROUND(AVG((RDATE - IDATE)), 2) AVGDAYS FROM LOAN WHERE  
RDATE IS NOT NULL
```

--Find the titles and resource types of the resources that are currently on loan,  
grouped by location.

```
SELECT NOTRETURN.RID, RESOURCEINFORMATION.TITLE,  
RESOURCEINFORMATION.RTYPE, NOTRETURN.LOCATION  
FROM (SELECT LOAN.RID, RESOURCES.ISBNEAN, RESOURCES.LOCATION  
FROM LOAN  
INNER JOIN RESOURCES ON LOAN.RID = RESOURCES.RID  
WHERE RDATE IS NULL) NOTRETURN  
INNER JOIN RESOURCEINFORMATION ON NOTRETURN.ISBNEAN =  
RESOURCEINFORMATION.ISBNEAN  
ORDER BY NOTRETURN.LOCATION
```

--Find the publisher name of the resource 'Database Systems'--

```
SELECT TITLE, ISBNEAN, PNAME  
FROM  
(SELECT RESOURCEINFORMATION.TITLE,  
RESOURCEINFORMATION.ISBNEAN, PUBLISHER.PNAME  
FROM RESOURCEINFORMATION  
INNER JOIN RESOURCEPUBLISHER  
ON RESOURCEINFORMATION.ISBNEAN = RESOURCEPUBLISHER.ISBNEAN  
INNER JOIN PUBLISHER  
ON RESOURCEPUBLISHER.PID = PUBLISHER.PID)  
WHERE TITLE = 'Database Systems';
```

--Count the number of loans taken out since 11/2020--

```
SELECT COUNT(IDATE) TOTALNOLOANS  
FROM LOAN  
WHERE (IDATE >= TO_DATE('01-11-2020', 'DD-MM-YYYY'));
```

--Find the list of titles for resources that were loaned out in November 2020 ordered alphabetically by title.--

```
SELECT DISTINCT TITLE, ISBNEAN
FROM RESOURCEINFORMATION
WHERE ISBNEAN IN
(SELECT ISBNEAN
FROM RESOURCES
JOIN LOAN
ON LOAN.RID = RESOURCES.RID
WHERE (IDATE <= TO_DATE('30-11-2020', 'DD-MM-YYYY')) AND (IDATE >=
TO_DATE('01-11-2020', 'DD-MM-YYYY')))
ORDER BY TITLE;
```

--Find the member name and ID of members who have a current valid membership with the library--

```
SELECT MNAME, ID
FROM MEMBERSHIP
WHERE (FD >= SYSDATE) AND SSTATUS = 'N';
```

--List the member names and IDs of members who have currently have a resource taken out with the library. --

```
SELECT DISTINCT MEMBERSHIP.MNAME, MEMBERSHIP.ID
FROM MEMBERSHIP
INNER JOIN
(SELECT EMAIL
FROM LOAN
WHERE LOAN.RDATE IS NULL) MWITHLOAN
ON MEMBERSHIP.EMAIL = MWITHLOAN.EMAIL;
```

--Find the titles, ISBNs, years of publication and edition of resources written by the author 'Linda Li', ordered by year of publication--

```
SELECT TITLE, ISBNEAN, PYEAR, EDITION
FROM
(SELECT RESOURCEINFORMATION.TITLE,
RESOURCEINFORMATION.ISBNEAN, RESOURCEINFORMATION.PYEAR,
RESOURCEINFORMATION.EDITION
```

```

FROM RESOURCEINFORMATION
INNER JOIN RESOURCEAUTHOR
ON RESOURCEINFORMATION.ISBNEAN = RESOURCEAUTHOR.ISBNEAN
INNER JOIN AUTHOR
ON RESOURCEAUTHOR.AID = AUTHOR.AID
WHERE ANAME = 'Linda Li')
ORDER BY PYEAR;

```

```

--List resource usage (history of a particular resource)
SELECT RESOURCEINFORMATION.TITLE, LOAN.EMAIL, LOAN.IDATE,
LOAN.RDATE FROM LOAN
INNER JOIN RESOURCES
ON LOAN.RID = RESOURCES.RID
INNER JOIN RESOURCEINFORMATION
ON RESOURCES.ISBNEAN = RESOURCEINFORMATION.ISBNEAN
WHERE RESOURCEINFORMATION.TITLE = 'Database Systems';

```

```

--Find the class numbers and names for all the titles loaned out since the
beginning of 2020, ordered by class number--
SELECT DISTINCT TITLE, CNUMBER, CNAME
FROM
(SELECT RESOURCEINFORMATION.TITLE, CLASSES.CNUMBER,
CLASSES.CNAME
FROM
CLASSES
INNER JOIN RESOURCEINFORMATION
ON RESOURCEINFORMATION.CNUMBER = CLASSES.CNUMBER
INNER JOIN RESOURCES
ON RESOURCEINFORMATION.ISBNEAN = RESOURCES.ISBNEAN
INNER JOIN LOAN
ON LOAN.RID = RESOURCES.RID
WHERE (LOAN.IDATE <= SYSDATE) AND (LOAN.IDATE >= TO_DATE('01-01-
2020', 'DD-MM-YYYY'))))
ORDER BY CNUMBER
;

```

```

--Show the number of resources a member is currently able to borrow
SELECT MEMBERSHIP.MNAME, MEMBERSHIP.EMAIL,
(CASE

```



```

WHEN SSTATUS = 'N' AND MEMBERSHIP.FD > SYSDATE THEN
NVL(TABLE100.RESOURCECANBORROW1, MEMBERSHIP.RCAP)
WHEN SSTATUS = 'Y' OR MEMBERSHIP.FD < SYSDATE THEN 0
END) AS RESOURCECANBORROW
FROM (SELECT LOAN.EMAIL,
MEMBERSHIP.RCAP - COUNT(CASE WHEN RDATE IS NULL THEN 1 END)
AS RESOURCECANBORROW1
FROM LOAN
INNER JOIN MEMBERSHIP
ON LOAN.EMAIL = MEMBERSHIP.EMAIL
GROUP BY LOAN.EMAIL, MEMBERSHIP.RCAP) TABLE100
FULL OUTER JOIN MEMBERSHIP
ON TABLE100.EMAIL = MEMBERSHIP.EMAIL;

```

### **Constraint Tests**

Here we will use constraints to verify our tables' integrity.

Constraints for MEMBERSHIP TABLE:

PK CONTRSAINT

INSERT INTO MEMBERSHIP

VALUES('michelle.emmanuel@college.ac.uk',10,'Emmanuel',1002340007,  
TO\_DATE('01-10-2017', 'DD-MM-YYYY'), TO\_DATE('01-12-2019', 'DD-MM-  
YYYY'), 'Staff', 'N');

SD < FD CONSTRAINT

INSERT INTO MEMBERSHIP

VALUES('Phu.Pham@college.ac.uk',10,'Phu',1024540007, TO\_DATE('01-10-  
2020', 'DD-MM-YYYY'), TO\_DATE('01-12-2019', 'DD-MM-YYYY'), 'Staff', 'N');

MTYPE CONSTRAINT

INSERT INTO MEMBERSHIP

VALUES('Phu.Pham@college.ac.uk',10,'Phu',1002322207, TO\_DATE('01-10-  
2017', 'DD-MM-YYYY'), TO\_DATE('01-12-2019', 'DD-MM-YYYY'), 'Boss', 'N');

EMAIL CONSTRAINT

INSERT INTO MEMBERSHIP

VALUES('Phu.Phamcollege.ac.uk',10,'Phu',1002322207, TO\_DATE('01-10-2017',  
'DD-MM-YYYY'), TO\_DATE('01-12-2019', 'DD-MM-YYYY'), 'Staff', 'N');

RCAP CONSTRAINT

```
INSERT INTO MEMBERSHIP
VALUES('Phu.Pham@college.ac.uk',12,'Phu',1002322207, TO_DATE('01-10-
2017', 'DD-MM-YYYY'), TO_DATE('01-12-2019', 'DD-MM-YYYY'), 'Staff', 'N');
```

MTYPE AND RCAP CONSTRAINT

```
INSERT INTO MEMBERSHIP
VALUES('Phu.Pham@college.ac.uk',10,'Phu',1002322207, TO_DATE('01-10-
2017', 'DD-MM-YYYY'), TO_DATE('01-12-2019', 'DD-MM-YYYY'), 'Student', 'N');
```

Constraints for CLASSES TABLE:

PK CONSTRAINT

```
INSERT INTO CLASSES VALUES(0001,'Finance');
```

Constraints for RESOURCEINFORMATION TABLE:

PK CONSTRAINT

```
INSERT INTO RESOURCEINFORMATION
VALUES(9781292061184,0021,'Book','New idea',7,'2012',14);
```

ISBNEAN CONSTRAINT

```
INSERT INTO RESOURCEINFORMATION
VALUES(978129207777761184,0001,'Book','Database Systems',6,'2010',14);
```

FK CONSTRAINT

```
INSERT INTO RESOURCEINFORMATION
VALUES(9781292041184,0022,'Book','Database Systems',6,'2010',14);
```

RTYPE CONSTRAINT

```
INSERT INTO RESOURCEINFORMATION
VALUES(9781117061184,0001,'Flier','Database Systems',6,'2010',14);
```

--Constraints to check the RESOURCES table foreign key. Inserting value for ISBNEAN that is not contained in referenced table (RESOURCEINFORMATION).

--

```
INSERT INTO RESOURCES (ISBNEAN, LOCATION, THCOPY) VALUES
(0123598235901, 0012, 3);
```

--Duplicate ISBNEAN- used to verify primary key.--

```
INSERT INTO RESOURCES (RID, ISBNEAN, LOCATION, THCOPY) VALUES
(2, 9780472001095, 0013, 4);
```

--Constraint to check the AUTHOR table and verify the primary key. 6 is already contained in the values for the primary key of this table.--

```
INSERT INTO AUTHOR VALUES (6, ' Sandi Toksvig ');
```

--Constraint to check the RESOURCEAUTHOR table composite foreign key ISBNEAN, AID. Inserting value ('15') for AID that is not contained in referenced table (AUTHOR). --

```
INSERT INTO RESOURCEAUTHOR VALUES (9781292061184, 15);
```

--Constraint to check the RESOURCEAUTHOR table composite foreign key ISBNEAN, AID. Inserting value for ISBNEAN that is not contained in referenced table (RESOURCEINFORMATION). --

```
INSERT INTO RESOURCEAUTHOR VALUES (9781292061100, 7);
```

-- Constraint to check the RESOURCEAUTHOR table composite primary key ISBNEAN, AID. Duplicate value for composite primary key ISBNEAN and AID used.--

```
INSERT INTO RESOURCEAUTHOR VALUES (9781292061184, 1);
```

--Constraint to check the length constraint on ISBNEAN by attempting to insert an ISBNEAN with 14 digits.

```
INSERT INTO RESOURCEAUTHOR VALUES (97812920611849, 1);
```

--Constraint to check the length constraint on ISBNEAN by attempting to insert an ISBNEAN with only 9 digits.

```
INSERT INTO RESOURCEAUTHOR VALUES (978129209, 1);
```

--Constraint to check the PUBLISHER table and verify the primary key. Attempting to insert row with primary key 'PID' ('3') already existing in the table.--

```
INSERT INTO PUBLISHER VALUES (3, 'Cambridge University Press');
```

-- repeated composite primary key, checking primary key --

```
INSERT INTO RESOURCEPUBLISHER VALUES (9781292061184, 1);
```

-- 9770000000000 is not in RESOURCEINFORMATION, checking foreign key--

```
INSERT INTO RESOURCEPUBLISHER VALUES (9770000000000,1);
```

-- 1996 is not in PID, checking foreign key--

```
INSERT INTO RESOURCEPUBLISHER VALUES (9770013061237,1996);
```

-- 977001306123732894858 is more than 13 digits, checking constraint for length--  
-

INSERT INTO RESOURCEPUBLISHER VALUES (977001306123732894858,1);

-- 97794858 is less than 10 digits, checking constraint for length--

INSERT INTO RESOURCEPUBLISHER VALUES (97794858,1);

-- repeated LREF, checking primary key--

INSERT INTO LOAN VALUES (3,'michelle.emmanuel@college.ac.uk', 1,  
TO\_DATE('21-11-2020', 'DD-MM-YYYY'), TO\_DATE('25-11-2020', 'DD-MM-  
YYYY'));

--email test.email@college.ac.uk is not in MEMBERSHIP, checking foreign key--

INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES  
(test.email@college.ac.uk', 1, TO\_DATE('01-09-2020', 'DD-MM-YYYY'),  
TO\_DATE('17-09-2020', 'DD-MM-YYYY'));

-- 1996 is not in RID, checking foreign key--

INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES  
(michelle.emmanuel@college.ac.uk', 1996, TO\_DATE('01-09-2020', 'DD-MM-  
YYYY'), TO\_DATE('17-09-2020', 'DD-MM-YYYY'));

-- IDATE after RDATE --

INSERT INTO LOAN (EMAIL,RID,IDATE,RDATE) VALUES  
(michelle.emmanuel@college.ac.uk', 1, TO\_DATE('01-12-2020', 'DD-MM-YYYY'),  
TO\_DATE('21-11-2020', 'DD-MM-YYYY'));

--repeated LREF, checking primary key--

INSERT INTO FINE VALUES (4, TO\_DATE('15-09-2020', 'DD-MM-YYYY'),  
'louis.edwards@college.ac.uk');

--1997 is not a LREF in LOAN, checking primary key and foreign key--

INSERT INTO FINE VALUES (1997, TO\_DATE('09-09-2020', 'DD-MM-YYYY'),  
'na021@college.ac.uk');

--test.email@college.ac.uk is not in MEMBERSHIP, checking foreign key--

INSERT INTO FINE VALUES (5, TO\_DATE('09-09-2020', 'DD-MM-YYYY'),  
'test.email@college.ac.uk');