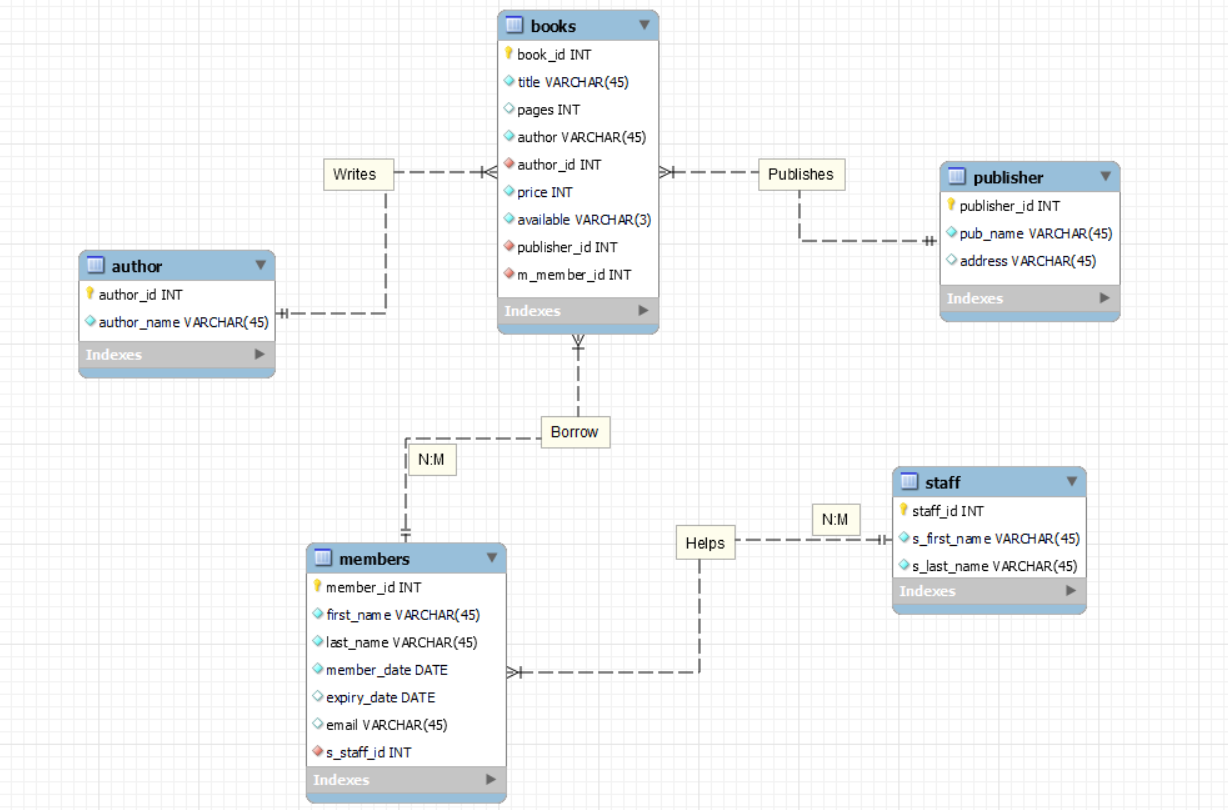
Student: Bakhtiyar Abdullayev

Group: BBA-041

Teacher: Javid Misirli

Library DBMS Project



**CREATING TABLES:**

CREATE TABLE author(

author\_id NUMBER,

author\_name VARCHAR(45) NOT NULL,

CONSTRAINT author\_const PRIMARY KEY(author\_id)

);

CREATE TABLE publisher(

publisher\_id NUMBER CONSTRAINT pub\_const PRIMARY KEY,

pub\_name VARCHAR(45) CONSTRAINT pb\_nm\_cns NOT NULL,

address VARCHAR(45)

);

CREATE TABLE books(

book\_id NUMBER PRIMARY KEY,

title VARCHAR(45) NOT NULL,

pages NUMBER,

author\_id NUMBER REFERENCES author(author\_id),

price NUMBER NOT NULL,

available VARCHAR(3) NOT NULL,

publisher\_id NUMBER REFERENCES publisher(publisher\_id)

);

CREATE TABLE staff(

staff\_id NUMBER PRIMARY KEY,

s\_first\_name VARCHAR(45) NOT NULL,

s\_last\_name VARCHAR(45) NOT NULL

);

CREATE TABLE members(

member\_id NUMBER,

first\_name VARCHAR(45) NOT NULL,

last\_name VARCHAR(45) NOT NULL,

member\_date DATE NOT NULL,

expiry\_date DATE,

email VARCHAR(45) UNIQUE

);

**INSERTING VALUES:**

INSERT INTO author

VALUES(1,'William Shakespeare');

INSERT INTO author

VALUES(2,'Agatha Christie');

INSERT INTO author

VALUES(3,'Leo Tolstoy');

INSERT INTO author

VALUES(4,'Fyodor Dostoevsky');

INSERT INTO author

VALUES(5,'Franz Kafka');

INSERT INTO publisher

VALUES(10,'Aspoligraf Ltd Mmc','Shahriyar st., 6 ');

INSERT INTO publisher

VALUES(20,'Azerneshr','M.Huseyn 61');

INSERT INTO publisher

VALUES(30,'Nafta-Press','H.Cavid pr. 29 A.');

INSERT INTO publisher

VALUES(40,'Chashioglu',null);

CREATE SEQUENCE books\_sqnc

INCREMENT BY 1

START WITH 1001

MAXVALUE 9999

NOCACHE

NOCYCLE;

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'Othello', 664, 1, 9.99, 'YES', 20);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'Men at War', 656, 3, 45, 'NO', 40);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'The Idiot', 215, 4, 25.5, 'NO', 40);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'Death on the Nile', 320, 2 ,20, 'YES', 10);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'Notes from Underground', 545, 4, 12.6, 'YES', 20);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'Romeo and Juliet', 951, 1, 45, 'NO', 30);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'Hamlet', 563, 1, 21.85, 'YES', 20);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'War & Peace', 314, 3, 21.1, 'YES', 10);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'Crime and Punishment', 654, 4, 65.3, 'YES', 10);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'The ABC Murders', 155, 2, 13, 'YES', 40);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'The Winter Tale', 544, 1, 45.6, 'NO', 20);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'Murder on the Orient Express', 203, 2, 12.08, 'YES', 40);

INSERT INTO books

VALUES(books\_sqnc.NEXTVAL, 'The Metamorphosis', 151, 5, 45.5, 'YES', 20);

CREATE SEQUENCE staff\_sqnc

INCREMENT BY 1

START WITH 10

MAXVALUE 9999

NOCACHE

NOCYCLE;

INSERT INTO staff

VALUES(staff\_sqnc.NEXTVAL, 'Adil', 'Tahmazov');

INSERT INTO staff

VALUES(staff\_sqnc.NEXTVAL, 'Riza', 'Alasgarov');

INSERT INTO staff

VALUES(staff\_sqnc.NEXTVAL, 'Eldar', 'Aliyev');

INSERT INTO staff

VALUES(staff\_sqnc.NEXTVAL, 'Zaur', 'Aghamaliyev');

INSERT INTO members

VALUES(110, 'Ashraf', 'Mammadov', '11-02-2020', null, 'ashrafmammadov@gmail.com');

INSERT INTO members

VALUES(111, 'Konul', 'Zeynalova', '20-08-2021', '31-12-2021', 'konul-z.29@mail.ru');

INSERT INTO members

VALUES(112,'Huseyn', 'Huseynli', '14-09-2020', null, 'hhuseynli20@gmail.com');

**CREATING VIEWS, SEQUENCES, SYNONYMS:**

CREATE VIEW project\_view

AS SELECT book\_id, title, price FROM books;

SELECT \* FROM project\_view;

CREATE VIEW projectview2

AS SELECT b.title, a.author\_name, b.price FROM books b

INNER JOIN author a

ON b.author\_id = a.author\_id;

SELECT \* FROM projectview2;

I have already created and used Sequences in inserting section.(colored them yellow)

CREATE SYNONYM project\_view\_copy

FOR project\_view;

**SUBQUERIES:**

For Example:

Choosing Agatha Christie’s books:

SELECT \* FROM books

WHERE author\_id = (SELECT author\_id FROM author WHERE author\_name='Agatha Christie');

Choosing books which was published by “Azerneshr”:

SELECT \* FROM books

WHERE publisher\_id = (SELECT publisher\_id FROM publisher WHERE pub\_name = 'Azerneshr');

Choosing the cheapest book:

SELECT book\_id, title, price

FROM books

WHERE price = (SELECT MIN(price) FROM books);

**USING ALL, ANY, IN:**

SELECT \* FROM books

WHERE author\_id = ANY(1,3,4);

(This means it will show us where the author\_id is equal 1,3 and 4(all of them))

We can write it like this, too:

SELECT \* FROM books

WHERE author\_id IN (1,3,4);

SELECT \* FROM books

WHERE pages<ALL(800,400,300);

(This means it will show us the books which has less pages than 800, 400 and 300. Basically we can write only the smallest number only(300))

**USING AND,OR:**

SELECT \* FROM books

WHERE author\_id = 4

OR

title = 'M%';

(It is okay if 1 condition is true)

SELECT \* FROM books

WHERE author\_id = 4

AND

pages>300;

(It will show us the results which compy with both conditions)

**JOINS:**

SELECT title, author\_name FROM books b

INNER JOIN author a

ON a.author\_id = b.author\_id;

SELECT \* FROM books

NATURAL JOIN publisher;

SELECT \* FROM books b, author a

WHERE

a.author\_id = b.author\_id;

SELECT \* FROM books b

RIGHT OUTER JOIN author a

ON a.author\_id = b.author\_id;

SELECT \* FROM books b

LEFT OUTER JOIN author a

ON a.author\_id = b.author\_id;

SELECT \* FROM books b

FULL OUTER JOIN author a

ON a.author\_id = b.author\_id;

SELECT \* FROM books

CROSS JOIN author;

SELECT \* FROM books

JOIN publisher

USING (publisher\_id);

**GROUP FUNCTIONS:**

SELECT MIN(pages)

FROM books;

SELECT MAX(price)

FROM books;

SELECT ROUND(AVG(price))

FROM books;

SELECT SUM(pages)

FROM books;

SELECT COUNT(\*)

FROM books

WHERE publisher\_id = 20;

SELECT COUNT(DISTINCT publisher\_id)

FROM books;

SELECT AVG(price)

FROM books

GROUP BY publisher\_id;

SELECT publisher\_id, SUM(pages)

FROM books

group by publisher\_id

HAVING SUM(pages)>1000;

**SINGLE ROW FUNCTIONS:**

SELECT (35\*553) FROM DUAL;

SELECT UPPER('baku') FROM DUAL;

SELECT LOWER('BAKU') FROM DUAL;

SELECT INITCAP('baku IS CApiTaL') FROM DUAL;

SELECT title, MOD(price, 2)

FROM books;

SELECT first\_name, last\_name, member\_date

FROM members

WHERE member\_date>to\_date('01-01-2020');

--how many months they are member of library

SELECT first\_name, last\_name, TRUNC((SYSDATE-member\_date)/30)

FROM members;

--using NVL

SELECT publisher\_id, pub\_name, NVL(address, 'Baku, Azerbaijan')

FROM publisher;

**RETRIEVING, RESTRICTING AND SORTING DATA:**

SELECT first\_name||' '||last\_name AS "Users"

FROM members;

SELECT DISTINCT publisher\_id

FROM books;

--where clause

SELECT \* FROM books

WHERE pages>500;

SELECT \* FROM books

WHERE price BETWEEN 20 AND 60;

SELECT \* FROM staff

WHERE s\_first\_name LIKE 'A%';

SELECT \* FROM books

FETCH FIRST 10 ROWS ONLY;

**CONSTRAINTS:**

I have already used constraints. Now I will add and drop them.

ALTER TABLE members

ADD CONSTRAINT mm\_cns PRIMARY KEY (member\_id);

ALTER TABLE publisher

DROP CONSTRAINT pb\_nm\_cns;

**DML AND DDL:**

--DML

UPDATE books

SET available = 'NO'

WHERE book\_id = 1003;

DELETE FROM books

WHERE book\_id = 1012;

--DDL

ALTER TABLE staff

ADD salary VARCHAR(20);

ALTER TABLE staff

MODIFY salary NUMBER;

ALTER TABLE staff

DROP COLUMN salary;

ALTER TABLE staff

RENAME COLUMN s\_first\_name

TO s\_f\_name;

ALTER TABLE staff

RENAME TO staff\_members;

DROP TABLE staff;

**References:**

[SQL Tutorial - GeeksforGeeks](https://www.geeksforgeeks.org/sql-tutorial/)

[SQL Tutorial - W3Schools](https://www.w3schools.com/sql/default.asp)