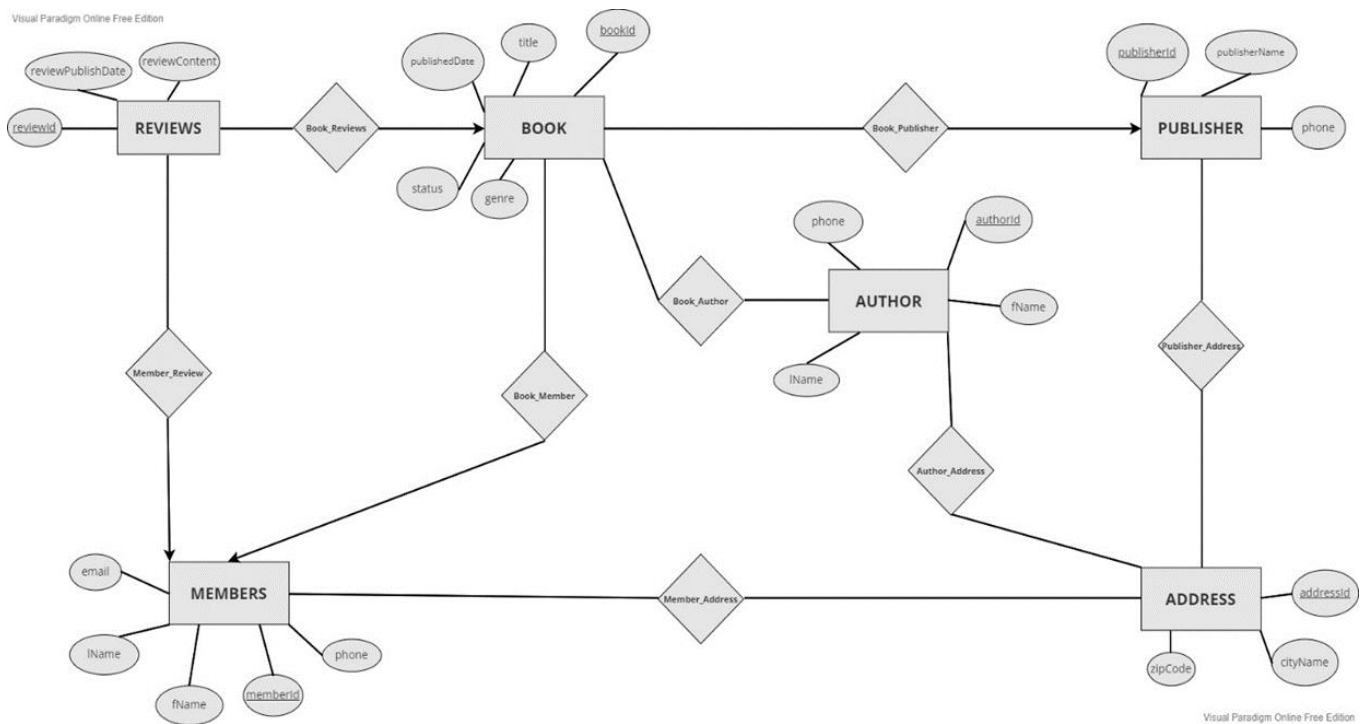


Library Management System

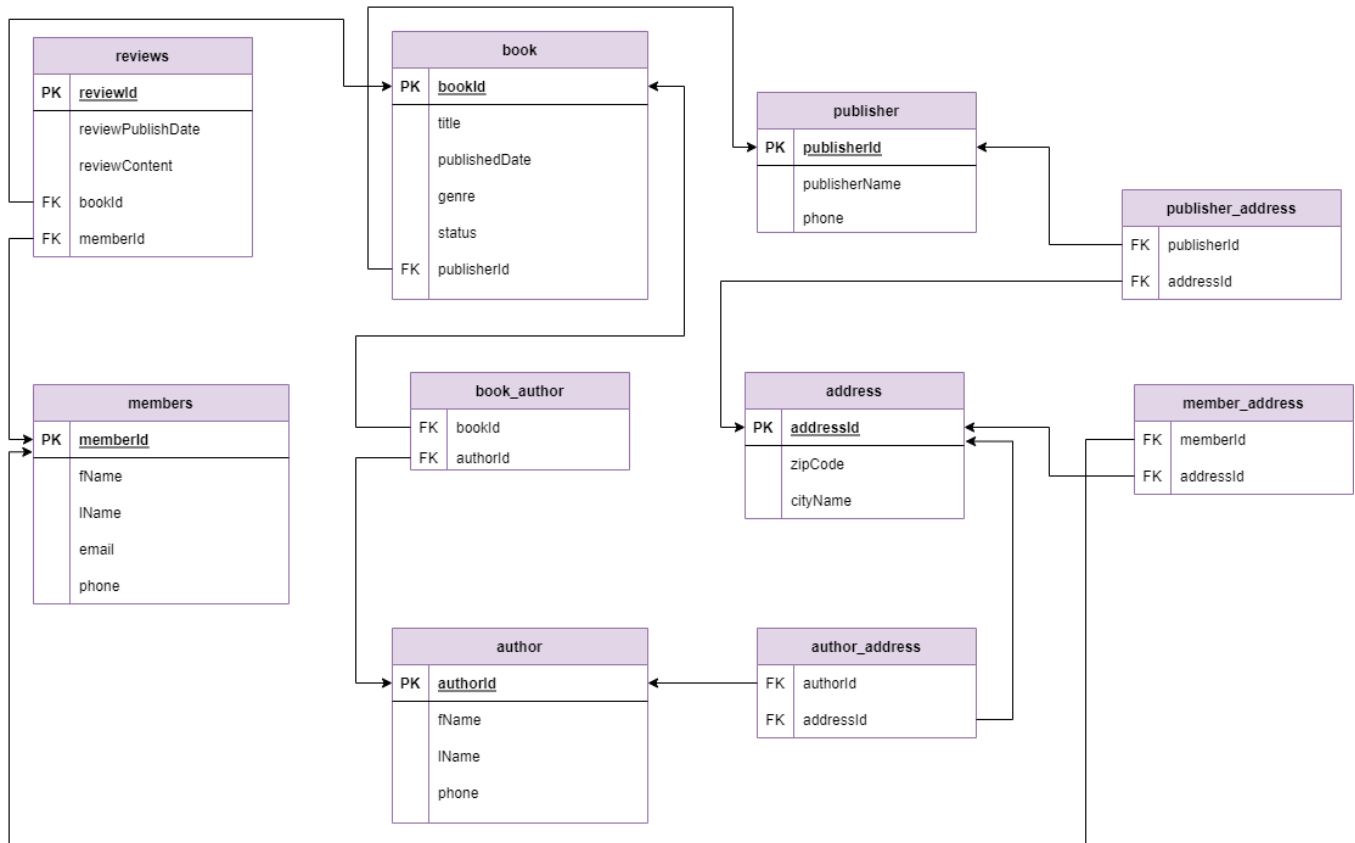
a. Topic and member-task responsibilities

In this project we designed a database management system that will be used in the library. Library Management System is a credible, efficient system for handling large amounts and multiple types of data which provides quick access to safe information. Library Management System provides smooth and organized working environment for the library management organization.

b. E-R Diagram



c. Database Schema



d. Create Table Statements

```
CREATE TABLE ADDRESS(  
    addressId NUMBER(4) PRIMARY KEY NOT NULL,  
    cityName VARCHAR2(20),  
    zipCode NUMBER(8)  
);
```

```
CREATE TABLE PUBLISHER(  
    publisherId NUMBER(4) PRIMARY KEY NOT NULL,  
    name VARCHAR2(20),  
    phone NUMBER(20)  
);
```

```
CREATE TABLE BOOK(  
    bookId NUMBER(4) PRIMARY KEY NOT NULL,  
    title VARCHAR2(100),  
    publishedDate DATE,  
    genre VARCHAR2(20),  
    status VARCHAR2(20),  
    publisherId NUMBER(4) FOREIGN KEY REFERENCES PUBLISHER(publisherId)
```

```
bookId NUMBER(4) PRIMARY KEY NOT NULL,  
title VARCHAR2(20),  
publishingDate DATE,  
status VARCHAR2(20),  
genra VARCHAR2(20),  
publisherId NUMBER(4) REFERENCES PUBLISHER(publisherId)  
);
```

```
CREATE TABLE AUTHORS(  
authorId NUMBER(4) PRIMARY KEY NOT NULL,  
fName VARCHAR2(20),  
lName VARCHAR2(20),  
phone NUMBER(20)  
);
```

```
CREATE TABLE MEMBERS(  
memberId NUMBER(4) PRIMARY KEY NOT NULL,  
fName VARCHAR2(20),  
lName VARCHAR2(20),  
phone NUMBER(20),  
email VARCHAR2(50),  
age NUMBER(2)  
);
```

```
CREATE TABLE REVIEWS(  
reviewId NUMBER(4) PRIMARY KEY,  
reviewContent VARCHAR2(180),  
reviewDate DATE,  
bookId NUMBER(4) REFERENCES BOOK(bookId),  
memberId NUMBER(4) REFERENCES MEMBERS(memberId)  
);
```

```

CREATE TABLE BOOK_AUTHOR(
    bookId NUMBER(4) REFERENCES BOOK(bookId),
    authorId NUMBER(4) REFERENCES AUTHORS(authorId),
    CONSTRAINT a_pk PRIMARY KEY ( bookId , authorId)
);

CREATE TABLE MEMBER_ADDRESS(
    addressId NUMBER(4) REFERENCES ADDRESS(addressId),
    memberId NUMBER(4) REFERENCES MEMBERS(memberId),
    CONSTRAINT b_pk PRIMARY KEY ( addressId , memberId)
);

CREATE TABLE PUBLISHER_ADDRESS(
    addressId NUMBER(4) REFERENCES ADDRESS(addressId),
    publisherId NUMBER(4) REFERENCES PUBLISHER(publisherId),
    CONSTRAINT c_pk PRIMARY KEY ( addressId , publisherId)
);

CREATE TABLE AUTHOR_ADDRESS(
    addressId NUMBER(4) REFERENCES ADDRESS(addressId),
    authorId NUMBER(4) REFERENCES AUTHORS(authorId),
    CONSTRAINT d_pk PRIMARY KEY ( addressId , authorId)
);

```

e. Insert Into Statements

1)Inserting for ADDRESS

```

INSERT INTO ADDRESS VALUES(06,'ANKARA',0606);
INSERT INTO ADDRESS VALUES(34,'ISTANBUL',3434);
INSERT INTO ADDRESS VALUES(38,'ISTANBUL',3438);
INSERT INTO ADDRESS VALUES(35,'IZMIR',3535);
INSERT INTO ADDRESS VALUES(50,'ANKARA',0650);
INSERT INTO ADDRESS VALUES(51,'ANTALYA',5156);
INSERT INTO ADDRESS VALUES(32,'MUGLA',3232);
INSERT INTO ADDRESS VALUES(48,'DENIZLI',4848);

```

Local.sql x Welcome Page x Local x

SQL Worksheet History

Worksheet Query Builder

```

INSERT INTO ADDRESS VALUES (06, 'ANKARA', 0606);
INSERT INTO ADDRESS VALUES (34, 'ISTANBUL', 3434);
INSERT INTO ADDRESS VALUES (38, 'ISTANBUL', 3438);
INSERT INTO ADDRESS VALUES (35, 'IZMIR', 3535);
INSERT INTO ADDRESS VALUES (50, 'ANKARA', 0650);
INSERT INTO ADDRESS VALUES (51, 'ANTALYA', 5156);
INSERT INTO ADDRESS VALUES (32, 'MUGLA', 3232);
INSERT INTO ADDRESS VALUES (48, 'DENIZLI', 4848);

SELECT * FROM ADDRESS;

```

Query... x

SQL All Rows Fetched: 8 in 0.334 seconds

	ADDRESSID	CITYNAME	ZIPCODE
1	6	ANKARA	606
2	34	ISTANBUL	3434
3	38	ISTANBUL	3438
4	35	IZMIR	3535
5	50	ANKARA	650
6	51	ANTALYA	5156
7	32	MUGLA	3232
8	48	DENIZLI	4848

2)Inserting for PUBLISHER

INSERT INTO PUBLISHER VALUES(1001,'KRONIK', 2522855001);

INSERT INTO PUBLISHER VALUES(1002, 'ITHAKI', 2522855002);

INSERT INTO PUBLISHER VALUES(1003, 'MENTIS', 2522855003);

INSERT INTO PUBLISHER VALUES(1004, 'ISBANKASI', 2522855004);

INSERT INTO PUBLISHER VALUES(1005, 'CAN', 2522855006);

INSERT INTO PUBLISHER VALUES(1006, 'YAPIKREDI',2522855005);

Local.sql x Welcome Page x Local x

SQL Worksheet History

Worksheet Query Builder

```

SELECT * FROM ADDRESS;

INSERT INTO PUBLISHER VALUES (1001, 'KRONIK', 2522855001);
INSERT INTO PUBLISHER VALUES (1002, 'ITHAKI', 2522855002);
INSERT INTO PUBLISHER VALUES (1003, 'MENTIS', 2522855003);
INSERT INTO PUBLISHER VALUES (1004, 'ISBANKASI', 2522855004);
INSERT INTO PUBLISHER VALUES (1005, 'CAN', 2522855006);
INSERT INTO PUBLISHER VALUES (1006, 'YAPIKREDI', 2522855005);

SELECT * FROM PUBLISHER;

```

Query Result x

All Rows Fetched: 6 in 0.063 seconds

	PUBLISHERID	NAME	PHONE
1	1001	KRONIK	2522855001
2	1002	ITHAKI	2522855002
3	1003	MENTIS	2522855003
4	1004	ISBANKASI	2522855004
5	1005	CAN	2522855006
6	1006	YAPIKREDI	2522855005

3)Inserting for BOOK

```

INSERT INTO BOOK VALUES (1111,'KASAGI' ,
TO_DATE('09/01/2000','DD/MM/YYYY'),'AVAILABLE','HORROR', 1001);

INSERT INTO BOOK VALUES (1211,'HACI KOMUNIST' ,
TO_DATE('20/12/2001','DD/MM/YYYY'),'AVAILABLE','COMEDY', 1002 );

INSERT INTO BOOK VALUES (3534,'ABDÜLHARIS PASA' ,
TO_DATE('11/02/2015','DD/MM/YYYY'),'BORROWED','FANTASY',1003 );

INSERT INTO BOOK VALUES (8954,'DUNE' ,
TO_DATE('02/03/1958','DD/MM/YYYY'),'AVAILABLE','SCI-FI', 1002 );

INSERT INTO BOOK VALUES (1254,'ZAMANIN KISA TARİH?' ,
TO_DATE('05/10/2003','DD/MM/YYYY'),'NOT AVAILABLE','SCIENCE',1004 );

INSERT INTO BOOK VALUES (8477,'MARS' ,
TO_DATE('09/03/2011','DD/MM/YYYY'),'BORROWED','COMEDY',1003 );

INSERT INTO BOOK VALUES (4695,'WATCHMEN' ,
TO_DATE('23/08/1978','DD/MM/YYYY'),'AVAILABLE','GRAPH NOVAL', 1006 );

INSERT INTO BOOK VALUES (1352,'SEFILLER' , TO_DATE('13/06/2000','DD/MM/YYYY'),'NOT
AVAILABLE', 'DRAM',1005 );

INSERT INTO BOOK VALUES (7413,'SUÇ VE CEZA' ,
TO_DATE('10/11/1887','DD/MM/YYYY'),'BORROWED','DRAM',1005 );

```

Local.sql x Welcome Page x Local x

SQL Worksheet | History

Worksheet | Query Builder

```

INSERT INTO BOOK VALUES (1111,'KASAGI' , TO_DATE('09/01/2000','DD/MM/YYYY'),'AVAILABLE','HORROR', 1001);
INSERT INTO BOOK VALUES (1211,'HACI KOMUNIST' , TO_DATE('20/12/2001','DD/MM/YYYY'),'AVAILABLE','COMEDY', 1002 );
INSERT INTO BOOK VALUES (3534,'ABDÜLHARIS PASA' , TO_DATE('11/02/2015','DD/MM/YYYY'),'BORROWED','FANTASY',1003 );
INSERT INTO BOOK VALUES (8954,'DUNE' , TO_DATE('02/03/1958','DD/MM/YYYY'),'AVAILABLE','SCI-FI', 1002 );
INSERT INTO BOOK VALUES (1254,'ZAMANIN KISA TARİH?' , TO_DATE('05/10/2003','DD/MM/YYYY'),'NOT AVAILABLE','SCIENCE',1004 );
INSERT INTO BOOK VALUES (8477,'MARS' , TO_DATE('09/03/2011','DD/MM/YYYY'),'BORROWED','COMEDY',1003 );
INSERT INTO BOOK VALUES (4695,'WATCHMEN' , TO_DATE('23/08/1978','DD/MM/YYYY'),'AVAILABLE','GRAPH NOVAL', 1006 );
INSERT INTO BOOK VALUES (1352,'SEFILLER' , TO_DATE('13/06/2000','DD/MM/YYYY'),'NOT AVAILABLE', 'DRAM',1005 );
INSERT INTO BOOK VALUES (7413,'SUÇ VE CEZA' , TO_DATE('10/11/1887','DD/MM/YYYY'),'BORROWED','DRAM',1005 );

SELECT * FROM BOOK;

```

Query Result x

All Rows Fetched: 9 in 0.031 seconds

	BOOKID	TITLE	PUBLISHINGDATE	STATUS	GENRA	PUBLISHERID
1	1111	KASAGI	09-JAN-00	AVAILABLE	HORROR	1001
2	1211	HACI KOMUNIST	20-DEC-01	AVAILABLE	COMEDY	1002
3	3534	ABDÜLHARIS PASA	11-FEB-15	BORROWED	FANTASY	1003
4	8954	DUNE	02-MAR-58	AVAILABLE	SCI-FI	1002
5	1254	ZAMANIN KISA...	05-OCT-03	NOT AVAILABLE	SCIENCE	1004
6	8477	MARS	09-MAR-11	BORROWED	COMEDY	1003
7	4695	WATCHMEN	23-AUG-78	AVAILABLE	GRAPH NOVAL	1006
8	1352	SEFILLER	13-JUN-00	NOT AVAILABLE	DRAM	1005
9	7413	SUC VE CEZA	10-NOV-87	BORROWED	DRAM	1005

4)Inserting for AUTHORS

```

INSERT INTO AUTHORS VALUES(001,'OMER', 'SEYFETT?N',5523403131);

INSERT INTO AUTHORS VALUES(002,'FERHAN', 'SENSOY',5523403132);

INSERT INTO AUTHORS VALUES(003,'MEHMET','YALTIRIK',5523403133);

INSERT INTO AUTHORS VALUES(004,'FRANK','HERBERT',5523403141);

INSERT INTO AUTHORS VALUES(006,'STEPHEN','HAWKING',5523403134);

INSERT INTO AUTHORS VALUES(008,'YASEMIN','CANSU',5523407131);

INSERT INTO AUTHORS VALUES(009,'ALAN','MOORE',5527403131);

INSERT INTO AUTHORS VALUES(010,'VICTOR','HUGO',5553403131);

INSERT INTO AUTHORS VALUES(011,'FYADOR','DOSTOYEVSKI',5523403139);

```

Local.sql x Welcome Page x Local x

SQL Worksheet History

Worksheet Query Builder

```

INSERT INTO AUTHORS VALUES (001, 'OMER', 'SEYFETT?N', 5523403131);
INSERT INTO AUTHORS VALUES (002, 'FERHAN', 'SENSOY', 5523403132);
INSERT INTO AUTHORS VALUES (003, 'MEHMET', 'YALTIRIK', 5523403133);
INSERT INTO AUTHORS VALUES (004, 'FRANK', 'HERBERT', 5523403141);
INSERT INTO AUTHORS VALUES (006, 'STEPHEN', 'HAWKING', 5523403134);
INSERT INTO AUTHORS VALUES (008, 'YASEMIN', 'CANSU', 5523407131);
INSERT INTO AUTHORS VALUES (009, 'ALAN', 'MOORE', 5527403131);
INSERT INTO AUTHORS VALUES (010, 'VICTOR', 'HUGO', 5553403131);
INSERT INTO AUTHORS VALUES (011, 'FYADOR', 'DOSTOYEVSKI', 5523403139);

SELECT * FROM AUTHORS;

```

Query Result x

All Rows Fetched: 9 in 0.055 seconds

	AUTHORID	FNAME	LNAME	PHONE
1	1	OMER	SEYFETT?N	5523403131
2	2	FERHAN	SENSOY	5523403132
3	3	MEHMET	YALTIRIK	5523403133
4	4	FRANK	HERBERT	5523403141
5	6	STEPHEN	HAWKING	5523403134
6	8	YASEMIN	CANSU	5523407131
7	9	ALAN	MOORE	5527403131
8	10	VICTOR	HUGO	5553403131
9	11	FYADOR	DOSTOYEVSKI	5523403139

5)Inserting for MEMBERS

INSERT INTO MEMBERS
VALUES(01,
'KAGAN'
, 'GENCER',

0123456789, 'kagan.gencer@bahcesehir.edu.tr', 21);

INSERT INTO MEMBERS VALUES(02, 'YASEMIN', 'KARACA', 0123456789,
'yasemin.karaca@bahcesehir.edu.tr', 33);

INSERT INTO MEMBERS VALUES(03, 'CANSU', 'KAZAZ', 1234657890,
'cansu.kazaz@bahcesehir.edu.tr', 12);

INSERT INTO MEMBERS VALUES(04, 'JOHN', 'DOE', 0123456789, 'john.doe@gmail', 85);

INSERT INTO MEMBERS VALUES(05, 'FATIH', 'ARSLAN', 0123456789, 'fatih.arslan@hotmail.com', 51
);

INSERT INTO MEMBERS VALUES(06, 'FERHAN', 'SENSOY', 0123456789,
'ferhan.sensoy@boun.edu.tr', 66);

Local.sql x Welcome Page x Local x

SQL Worksheet History

Worksheet Query Builder

```

INSERT INTO MEMBERS VALUES(01, 'KAGAN' , 'GENCER', 0123456789, 'kagan.gencer@bahcesehir.edu.tr', 21);
INSERT INTO MEMBERS VALUES(02, 'YASEMIN' , 'KARACA', 0123456789, 'yasemin.karaca@bahcesehir.edu.tr', 33 );
INSERT INTO MEMBERS VALUES(03, 'CANSU' , 'KAZAZ', 1234657890, 'cansu.kazaz@bahcesehir.edu.tr', 12 );
INSERT INTO MEMBERS VALUES(04, 'JOHN' , 'DOE', 0123456789, 'john.doe@gmail', 85 );
INSERT INTO MEMBERS VALUES(05, 'FATIH' , 'ARSLAN', 0123456789, 'fatih.arслан@hotmail.com', 51 );
INSERT INTO MEMBERS VALUES(06, 'FERHAN' , 'SENSOY', 0123456789, 'ferhan.sensoy@boun.edu.tr', 66 );

SELECT * FROM MEMBERS;

INSERT INTO BOOK_AUTHOR VALUES(1111,001);

```

Query Result x

All Rows Fetched: 6 in 0.049 seconds

	MEMBERID	FNAME	LNAME	PHONE	EMAIL	AGE
1	1	KAGAN	GENCER	123456789	kagan.gencer@bahcesehir.edu.tr	21
2	2	YASEMIN	KARACA	123456789	yasemin.karaca@bahcesehir.edu.tr	33
3	3	CANSU	KAZAZ	1234657890	cansu.kazaz@bahcesehir.edu.tr	12
4	4	JOHN	DOE	123456789	john.doe@gmail	85
5	5	FATIH	ARSLAN	123456789	fatih.arслан@hotmail.com	51
6	6	FERHAN	SENSOY	123456789	ferhan.sensoy@boun.edu.tr	66

6)Inserting for BOOK_AUTHOR

INSERT INTO BOOK_AUTHOR VALUES(1111,001);

INSERT INTO BOOK_AUTHOR VALUES(1211,002);

INSERT INTO BOOK_AUTHOR VALUES(3534,003);

INSERT INTO BOOK_AUTHOR VALUES(8954,004);

INSERT INTO BOOK_AUTHOR VALUES(1254,006);

INSERT INTO BOOK_AUTHOR VALUES(8477,007);

INSERT INTO BOOK_AUTHOR VALUES(4695,008);

INSERT INTO BOOK_AUTHOR VALUES(1352,009);

INSERT INTO BOOK_AUTHOR VALUES(7413,010);

The screenshot shows a SQL Worksheet interface with two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying a series of SQL INSERT statements for the BOOK_AUTHOR table, followed by a SELECT statement. Below the queries, the 'Query Result' tab shows the results of the SELECT statement, displaying 8 rows of data with columns BOOKID and AUTHORID.

SQL Queries:

```

INSERT INTO BOOK_AUTHOR VALUES (1111, 001);
INSERT INTO BOOK_AUTHOR VALUES (1211, 002);
INSERT INTO BOOK_AUTHOR VALUES (3534, 003);
INSERT INTO BOOK_AUTHOR VALUES (8954, 004);
INSERT INTO BOOK_AUTHOR VALUES (1254, 006);
INSERT INTO BOOK_AUTHOR VALUES (8477, 007);
INSERT INTO BOOK_AUTHOR VALUES (4695, 008);
INSERT INTO BOOK_AUTHOR VALUES (1352, 009);
INSERT INTO BOOK_AUTHOR VALUES (7413, 010);

SELECT * FROM BOOK_AUTHOR;

```

Query Result: All Rows Fetched: 8 in 4.001 seconds

	BOOKID	AUTHORID
1	1111	1
2	1211	2
3	3534	3
4	8954	4
5	1254	6
6	4695	8
7	1352	9
8	7413	10

7) Inserting for AUTHOR_ADDRESS

```

INSERT INTO AUTHOR_ADDRESS VALUES(35,001);
INSERT INTO AUTHOR_ADDRESS VALUES(34,002);
INSERT INTO AUTHOR_ADDRESS VALUES(38,003);
INSERT INTO AUTHOR_ADDRESS VALUES(06,004);
INSERT INTO AUTHOR_ADDRESS VALUES(48,006);
INSERT INTO AUTHOR_ADDRESS VALUES(34,008);
INSERT INTO AUTHOR_ADDRESS VALUES(32,009);
INSERT INTO AUTHOR_ADDRESS VALUES(50,010);
INSERT INTO AUTHOR_ADDRESS VALUES(51,011);

```

Local.sql x Welcome Page x Local x

SQL Worksheet History

Worksheet Query Builder

```

INSERT INTO AUTHOR_ADDRESS VALUES (35,001);
INSERT INTO AUTHOR_ADDRESS VALUES (34,002);
INSERT INTO AUTHOR_ADDRESS VALUES (38,003);
INSERT INTO AUTHOR_ADDRESS VALUES (06,004);
INSERT INTO AUTHOR_ADDRESS VALUES (48,006);
INSERT INTO AUTHOR_ADDRESS VALUES (34,008);
INSERT INTO AUTHOR_ADDRESS VALUES (32,009);
INSERT INTO AUTHOR_ADDRESS VALUES (50,010);
INSERT INTO AUTHOR_ADDRESS VALUES (51,011);

SELECT * FROM AUTHOR_ADDRESS;

```

Query Result x

All Rows Fetched: 9 in 0.037 seconds

	ADDRESSID	AUTHORID
1	35	1
2	34	2
3	38	3
4	6	4
5	48	6
6	34	8
7	32	9
8	50	10
9	51	11

8)Inserting for MEMBER_ADDRESS

```

INSERT INTO MEMBER_ADDRESS VALUES(34,001);
INSERT INTO MEMBER_ADDRESS VALUES(34,002);
INSERT INTO MEMBER_ADDRESS VALUES(06,003);
INSERT INTO MEMBER_ADDRESS VALUES(06,004);
INSERT INTO MEMBER_ADDRESS VALUES(48,005);
INSERT INTO MEMBER_ADDRESS VALUES(48,006);

```

The screenshot shows the SQL Developer interface. The top pane contains two SQL queries. The first query consists of six INSERT statements into the MEMBER_ADDRESS table. The second query is a SELECT statement that retrieves all data from the MEMBER_ADDRESS table. The bottom pane displays the query results for the SELECT statement, showing a table with 6 rows and 2 columns: ADDRESSID and MEMBERID.

```

INSERT INTO MEMBER_ADDRESS VALUES (34,001);
INSERT INTO MEMBER_ADDRESS VALUES (34,002);
INSERT INTO MEMBER_ADDRESS VALUES (06,003);
INSERT INTO MEMBER_ADDRESS VALUES (06,004);
INSERT INTO MEMBER_ADDRESS VALUES (48,005);
INSERT INTO MEMBER_ADDRESS VALUES (48,006);

SELECT * FROM MEMBER_ADDRESS;

```

	ADDRESSID	MEMBERID
1	34	1
2	34	2
3	6	3
4	6	4
5	48	5
6	48	6

9)Inserting for PUBLISHER_ADDRESS

```

INSERT INTO PUBLISHER_ADDRESS VALUES(34,1001);
INSERT INTO PUBLISHER_ADDRESS VALUES(34,1002);
INSERT INTO PUBLISHER_ADDRESS VALUES(06,1003);
INSERT INTO PUBLISHER_ADDRESS VALUES(06,1004);
INSERT INTO PUBLISHER_ADDRESS VALUES(38,1005);
INSERT INTO PUBLISHER_ADDRESS VALUES(48,1006);

```

The screenshot shows a SQL worksheet with the following query:

```

INSERT INTO PUBLISHER_ADDRESS VALUES (34,1001);
INSERT INTO PUBLISHER_ADDRESS VALUES (34,1002);
INSERT INTO PUBLISHER_ADDRESS VALUES (06,1003);
INSERT INTO PUBLISHER_ADDRESS VALUES (06,1004);
INSERT INTO PUBLISHER_ADDRESS VALUES (38,1005);
INSERT INTO PUBLISHER_ADDRESS VALUES (48,1006);

SELECT * FROM PUBLISHER_ADDRESS;

```

The results of the query are displayed in a table below the query:

	ADDRESSID	PUBLISHERID
1	34	1001
2	34	1002
3	6	1003
4	6	1004
5	38	1005
6	48	1006

10) Inserting for REVIEWS

INSERT INTO REVIEWS VALUES(1001,'it is epic book ever in the history',TO_DATE('06/11/2000','DD/MM/YYYY'),1111,0001);

INSERT INTO REVIEWS VALUES(1002, 'I love it so much', TO_DATE('11/02/2001','DD/MM/YYYY'),1111,0002);

INSERT INTO REVIEWS VALUES(1003, 'I hate this.', TO_DATE('11/02/2011','DD/MM/YYYY'),1254,0002);

INSERT INTO REVIEWS VALUES(1004, 'it was hillirous',TO_DATE('08/08/2008','DD/MM/YYYY'),7413,0003);

INSERT INTO REVIEWS VALUES(1005, 'I cried a lot',TO_DATE('01/01/2001','DD/MM/YYYY'),1352,0004);

INSERT INTO REVIEWS VALUES(1006, 'who watch the WATCHMEN',TO_DATE('02/04/1999','DD/MM/YYYY'),4695,0004);

INSERT INTO REVIEWS VALUES(1007, 'Everyone should read this mate!!!',TO_DATE('02/05/1999','DD/MM/YYYY'),4695,0005);

The screenshot shows the SQL Developer interface. The top window is the 'SQL Worksheet' with the following SQL code:

```

INSERT INTO REVIEWS VALUES(1001,'it is epic book ever in the history',TO_DATE('06/11/2000','DD/MM/YYYY'),1111,0001);
INSERT INTO REVIEWS VALUES(1002,'I love it so much', TO_DATE('11/02/2001','DD/MM/YYYY'),1111,0002);
INSERT INTO REVIEWS VALUES(1003,'I hate this.', TO_DATE('11/02/2011','DD/MM/YYYY') ,1254,0002);
INSERT INTO REVIEWS VALUES(1004,'it was hillirous',TO_DATE('08/08/2008','DD/MM/YYYY'),7413,0003);
INSERT INTO REVIEWS VALUES(1005,'I cried a lot',TO_DATE('01/01/2001','DD/MM/YYYY'), 1352,0004);
INSERT INTO REVIEWS VALUES(1006,'who watch the WATCHMEN',TO_DATE('02/04/1999','DD/MM/YYYY'),4695,0004);
INSERT INTO REVIEWS VALUES(1007,'Everyone should read this mate!!!',TO_DATE('02/05/1999','DD/MM/YYYY'),4695,0005);

SELECT * FROM REVIEWS;

```

The bottom window is the 'Query Result' showing the results of the SELECT statement. It displays 7 rows of data with columns: REVIEWID, REVIEWCONTENT, REVIEWDATE, BOOKID, and MEMBERID.

REVIEWID	REVIEWCONTENT	REVIEWDATE	BOOKID	MEMBERID
1	1001it is epic book ever in the history	06-NOV-00	1111	1
2	1002I love it so much	11-FEB-01	1111	2
3	1003I hate this.	11-FEB-11	1254	2
4	1004it was hillirous	08-AUG-08	7413	3
5	1005I cried a lot	01-JAN-01	1352	4
6	1006who watch the WATCHMEN	02-APR-99	4695	4
7	1007Everyone should read this mate!!!	02-MAY-99	4695	5

f. SQL Queries

- 2 Joins (with conditions)

1) Find the title of the books and their publisher's name&ID

SELECT BOOK.publisherId, BOOK.title, PUBLISHER.name

FROM BOOK JOIN PUBLISHER ON BOOK.publisherId=PUBLISHER.publisherId;

The screenshot shows the SQL Developer interface. The top window is the 'SQL Worksheet' with the following SQL code:

```

/*JOINS*/
/*Find the title of books and publisher name .*/

SELECT BOOK.publisherId, BOOK.title, PUBLISHER.name
FROM BOOK JOIN PUBLISHER ON BOOK.publisherId=PUBLISHER.publisherId;

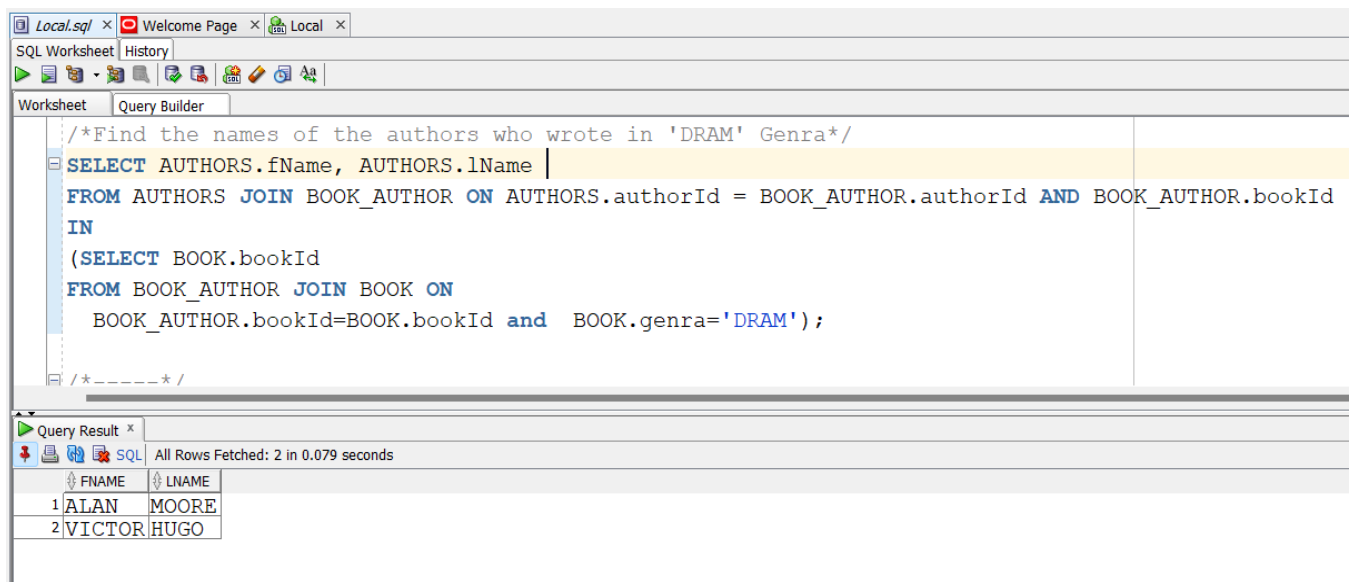
```

The bottom window is the 'Query Result' showing the results of the JOIN query. It displays 9 rows of data with columns: PUBLISHERID, TITLE, and NAME.

PUBLISHERID	TITLE	NAME
1	1001KASAGI	KRONIK
2	1002HACI KOMUNIST	ITHAKI
3	1003ABDÜLHARIS PASA	MENTIS
4	1002DUNE	ITHAKI
5	1004ZAMANIN KISA TARİH?	ISBANKASI
6	1003MARS	MENTIS
7	1006WATCHMEN	YAPIKREDİ
8	1005SEFİLLER	CAN
9	1005SUÇ VE CEZA	CAN

2) Find the names of the authors who wrote in 'DRAM' genre

```
SELECT AUTHORS.fName, AUTHORS.lName  
  
FROM AUTHORS JOIN BOOK_AUTHOR ON AUTHORS.authorId = BOOK_AUTHOR.authorId  
AND BOOK_AUTHOR.bookId  
  
IN  
  
(SELECT BOOK.bookId  
  
FROM BOOK_AUTHOR JOIN BOOK ON  
  
BOOK_AUTHOR.bookId=BOOK.bookId and BOOK.genra='DRAM');
```



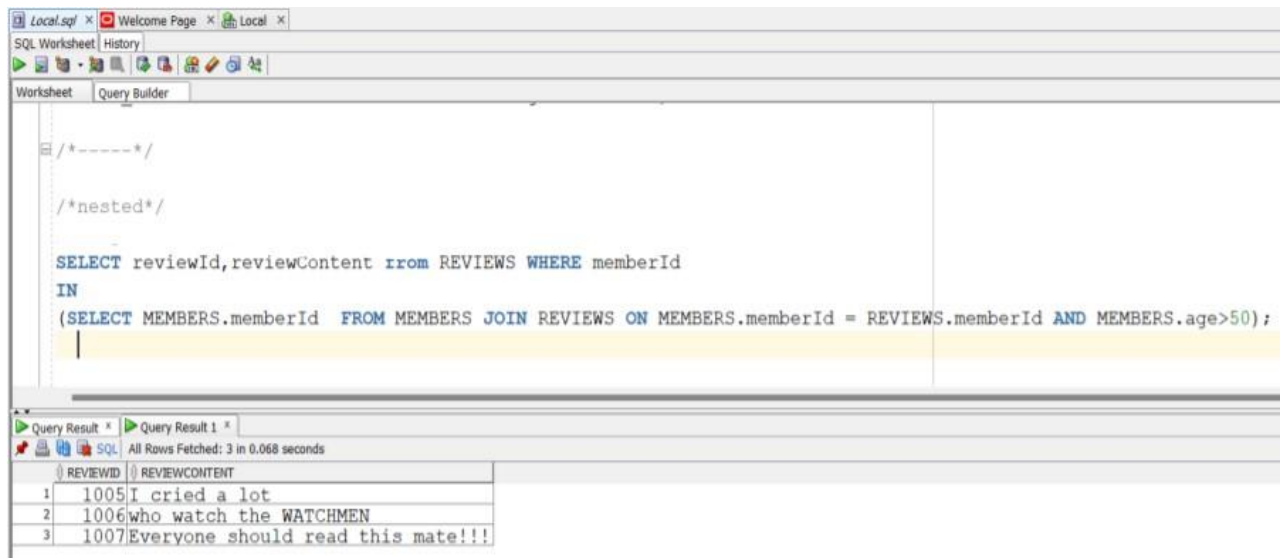
The screenshot shows an SQL IDE interface. The top pane displays a query with a comment: `/*Find the names of the authors who wrote in 'DRAM' Genra*/`. The query is: `SELECT AUTHORS.fName, AUTHORS.lName FROM AUTHORS JOIN BOOK_AUTHOR ON AUTHORS.authorId = BOOK_AUTHOR.authorId AND BOOK_AUTHOR.bookId IN (SELECT BOOK.bookId FROM BOOK_AUTHOR JOIN BOOK ON BOOK_AUTHOR.bookId=BOOK.bookId and BOOK.genra='DRAM');`. The bottom pane shows the query results in a table with two columns, FNAME and LNAME. The results are: 1 | ALAN | MOORE and 2 | VICTOR | HUGO.

	FNAME	LNAME
1	ALAN	MOORE
2	VICTOR	HUGO

- 2 Nested Queries

1) Show IDs and reviews of the members who are over 50

```
SELECT reviewId,reviewContent from REVIEWS WHERE memberId  
  
IN  
  
(SELECT MEMBERS.memberId FROM MEMBERS JOIN REVIEWS  
  
ON  
  
MEMBERS.memberId = REVIEWS.memberId AND MEMBERS.age>50);
```



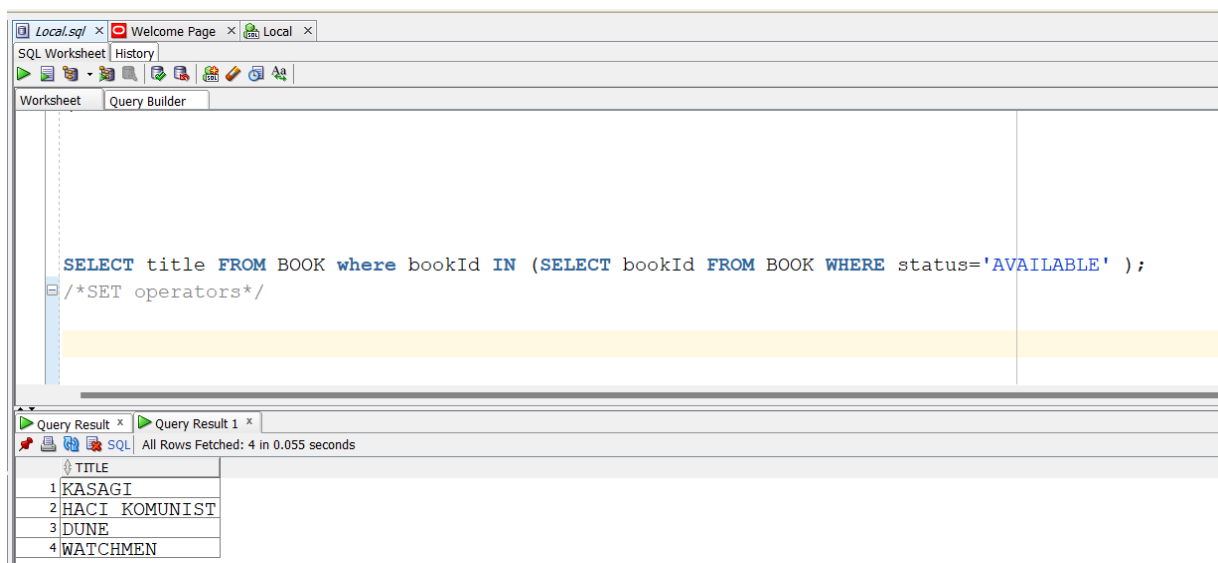
The screenshot shows a SQL Worksheet interface with a query editor and a results pane. The query is a nested SELECT statement. The results pane shows three rows of data.

```
/*-----*/  
  
/*nested*/  
  
SELECT reviewId,reviewContent from REVIEWS WHERE memberId  
IN  
(SELECT MEMBERS.memberId FROM MEMBERS JOIN REVIEWS ON MEMBERS.memberId = REVIEWS.memberId AND MEMBERS.age>50);
```

REVIEWID	REVIEWCONTENT
1005	I cried a lot
1006	who watch the WATCHMEN
1007	Everyone should read this mate!!!

2) Find the books whose status are 'AVAILABLE'

```
SELECT title FROM BOOK WHERE bookId  
  
IN  
  
(SELECT bookId FROM BOOK WHERE status='AVAILABLE' );
```



The screenshot shows a SQL Worksheet interface with a query editor and a results pane. The query is a nested SELECT statement. The results pane shows four rows of data.

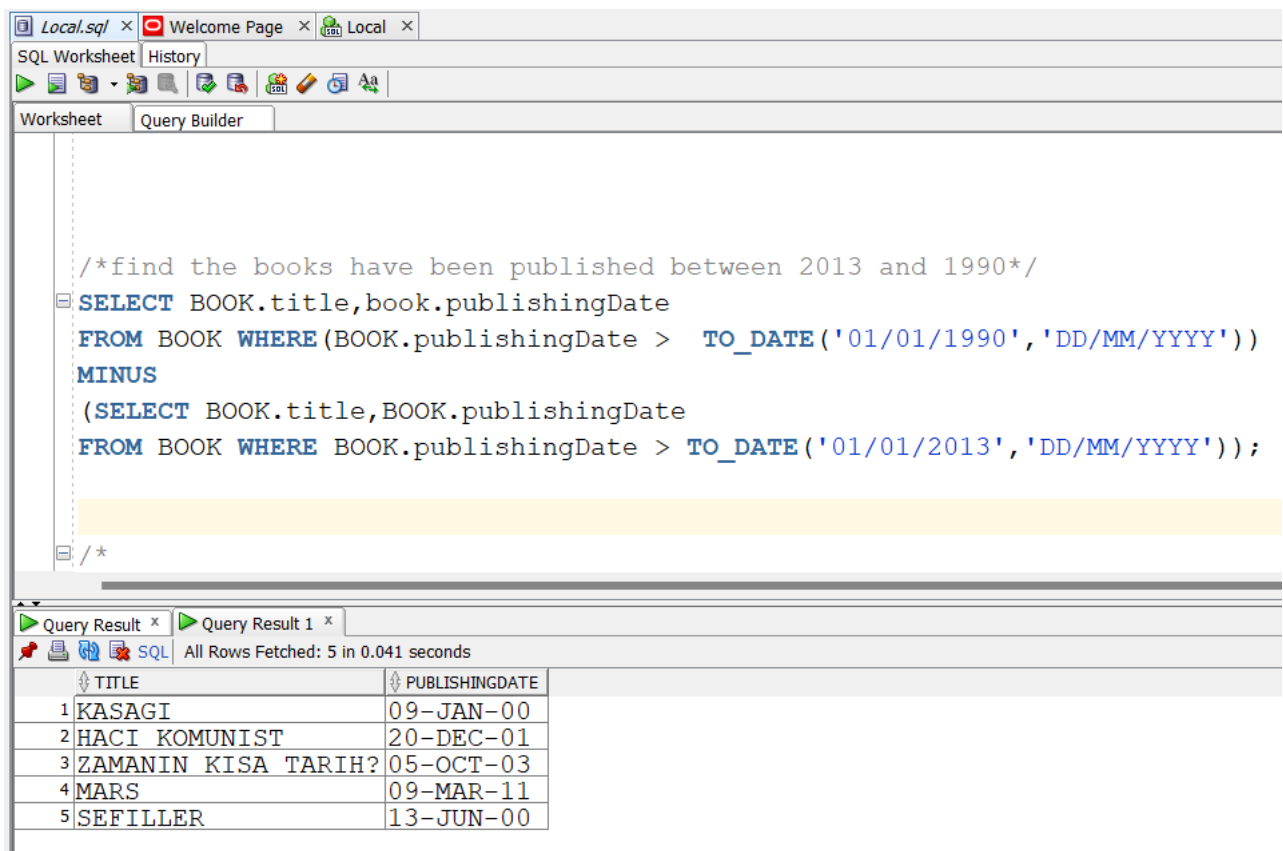
```
SELECT title FROM BOOK where bookId IN (SELECT bookId FROM BOOK WHERE status='AVAILABLE' );  
/*SET operators*/
```

TITLE
1 KASAGI
2 HADI KOMUNIST
3 DUNE
4 WATCHMEN

- 2 Set Operations

1) Find the books which have been published between 2013 and 1990

```
SELECT BOOK.title,book.publishingDate
FROM BOOK WHERE(BOOK.publishingDate > TO_DATE('01/01/1990','DD/MM/YYYY'))
MINUS
(SELECT BOOK.title,BOOK.publishingDate
FROM BOOK WHERE BOOK.publishingDate > TO_DATE('01/01/2013','DD/MM/YYYY'));
```



The screenshot shows an SQL IDE window with a query editor and a results pane. The query editor contains the following SQL code:

```
/*find the books have been published between 2013 and 1990*/
SELECT BOOK.title,book.publishingDate
FROM BOOK WHERE(BOOK.publishingDate > TO_DATE('01/01/1990','DD/MM/YYYY'))
MINUS
(SELECT BOOK.title,BOOK.publishingDate
FROM BOOK WHERE BOOK.publishingDate > TO_DATE('01/01/2013','DD/MM/YYYY'));
```

The results pane shows the following data:

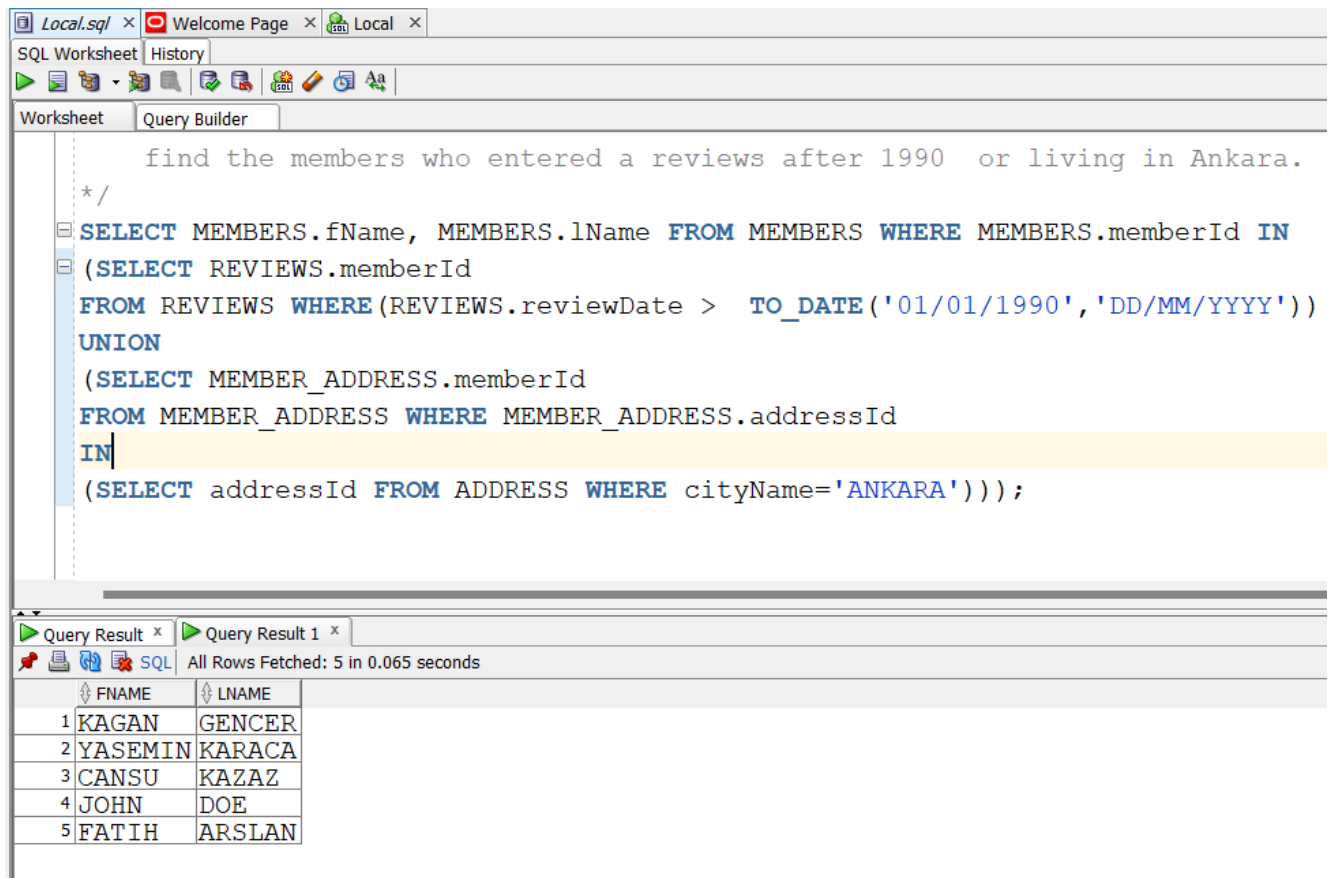
	TITLE	PUBLISHINGDATE
1	KASAGI	09-JAN-00
2	HACI KOMUNIST	20-DEC-01
3	ZAMANIN KISA TARİH?	05-OCT-03
4	MARS	09-MAR-11
5	SEFILLER	13-JUN-00

2) Find the members who wrote a review after 1990 or living in Ankara

```
SELECT MEMBERS.fName, MEMBERS.lName FROM MEMBERS WHERE MEMBERS.memberId
IN
(SELECT REVIEWS.memberId
FROM REVIEWS WHERE(REVIEWS.reviewDate > TO_DATE('01/01/1990','DD/MM/YYYY'))
UNION
(SELECT MEMBER_ADDRESS.memberId
FROM MEMBER_ADDRESS WHERE MEMBER_ADDRESS.addressId
```

IN

```
(SELECT addressId FROM ADDRESS WHERE cityName='ANKARA')));
```



The screenshot shows an SQL IDE with a query editor and a results pane. The query editor contains a comment and an SQL query. The results pane shows a table with 5 rows and 2 columns: FNAME and LNAME.

```
find the members who entered a reviews after 1990 or living in Ankara.  
*/  
SELECT MEMBERS.fName, MEMBERS.lName FROM MEMBERS WHERE MEMBERS.memberId IN  
(SELECT REVIEWS.memberId  
FROM REVIEWS WHERE (REVIEWS.reviewDate > TO_DATE('01/01/1990', 'DD/MM/YYYY'))  
UNION  
(SELECT MEMBER_ADDRESS.memberId  
FROM MEMBER_ADDRESS WHERE MEMBER_ADDRESS.addressId  
IN  
(SELECT addressId FROM ADDRESS WHERE cityName='ANKARA')));
```

Query Result 1 x
All Rows Fetched: 5 in 0.065 seconds

	FNAME	LNAME
1	KAGAN	GENCER
2	YASEMIN	KARACA
3	CANSU	KAZAZ
4	JOHN	DOE
5	FATIH	ARSLAN

- 2 Aggregate Operations

1) Find the number of the members who live in the same city

```
SELECT addressId, COUNT(memberId)
```

```
FROM
```

```
(SELECT member_address.addressId, member_address.memberId ,address.cityName
```

```
FROM member_address JOIN address ON member_address.addressId=address.addressId)
```

```
GROUP BY addressId;
```

Local.sql x Welcome Page x Local x

SQL Worksheet History

Worksheet Query Builder

```

/* Aggregator*/

/* Count of members in the same city */
SELECT addressId, COUNT(memberId)
FROM (SELECT member_address.addressId, member_address.memberId ,address.cityName
FROM member_address JOIN address ON
member_address.addressId=address.addressId) GROUP BY addressId;

```

Query Result x Query Result 1 x

All Rows Fetched: 3 in 0.047 seconds

	ADDRESSID	COUNT(MEMBERID)
1	6	2
2	34	2
3	48	2

2) Find the average age of the members who wrote a review

SELECT AVG(age) FROM MEMBERS JOIN REVIEWS ON MEMBERS.memberId = REVIEWS.memberId;

Local.sql x Welcome Page x Local x

SQL Worksheet History

Worksheet Query Builder

```

SELECT AVG(age) FROM MEMBERS JOIN REVIEWS ON MEMBERS.memberId = REVIEWS.memberId;

```

Query Result x Query Result 1 x

All Rows Fetched: 1 in 0.031 seconds

	AVG(AGE)
1	45.71428571428571428571428571428571