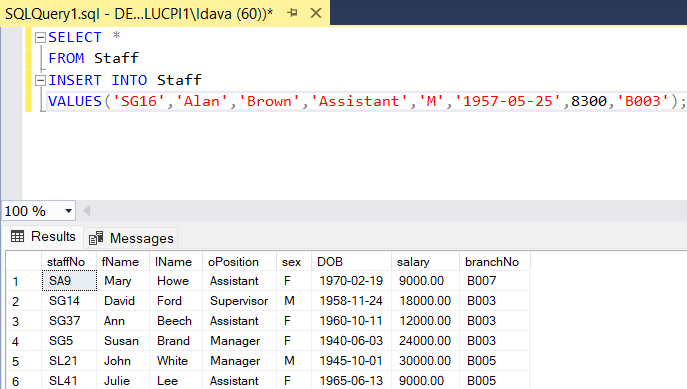
*Дасгал 1. Лекцийн хичээлдээр үзсэн жишээ асуулгуудыг бичиж ажиллуулж үр дүнг харах. Жишээ 5.35 -5.42 хүртэл бичиж ажиллуулж үзэх. (Ажлуулсан үр дүнг бичсэн тайланг PDF хэлбрээр явуулна.)*

*5.35*

*MSSQL*

**

*MYSQL*

*Graphical user interface, application

Description automatically generated*

*PostgreSQL*

*Graphical user interface, application

Description automatically generated*

*5.36*

*MSSQL:*

*Table

Description automatically generated*

*MYSQL:*

*Graphical user interface, application, email

Description automatically generated*

*PostgreSQL*

*Graphical user interface

Description automatically generated*

*5.37*

*MSSQL:*

*Text

Description automatically generated*

*MYSQL:*

*Graphical user interface, text, application, email

Description automatically generated*

*PostgreSQL:*

*Text

Description automatically generated with medium confidence*

*5.38*

*SELECT \**

*FROM Staff*

*UPTADE Staff;*

*SET salary = salary\*1.03;*

*SELECT \**

*FROM Staff*

*MSSQL*

*Graphical user interface, table

Description automatically generated with medium confidence*

*MYSQL:*

*Graphical user interface, application

Description automatically generated*

*PostgreSQL:*

*Table

Description automatically generated*

*5.39*

*UPTADE Staff*

*SET salary = salary\*1.05*

*WHERE oPosition = 'Manager';*

*SELECT \**

*FROM Staff*

*WHERE oPosition = 'Manager'*

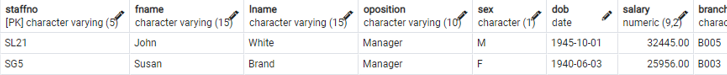
*MSSQL:*

*Graphical user interface

Description automatically generated with medium confidence  
MYSQL:*

*Graphical user interface, application, Teams

Description automatically generated  
PostgreSQL:*

**

*5.40*

*UPTADE Staff*

*SET oPosition = 'Manager', salary = 18000*

*WHERE staffNo = 'SG14'*

*SELECT \**

*FROM Staff*

*WHERE staffNo = 'SG14'*

*MSSQL:*

**

*MYSQL:*

**

*PostgreSQL:*

**

*5.41*

*SELECT \**

*FROM Viewing*

*DELETE propertyNo = 'PG4'*

*SELECT \**

*FROM Viewing*

*MSSQL:*

*Table

Description automatically generated  
MYSQL:*

*Table

Description automatically generated  
PostgreSQL:*

*Table

Description automatically generated*

*5.42*

*MSSQL:*

*Graphical user interface, text, application

Description automatically generated  
MYSQL:*

*Graphical user interface, application

Description automatically generated  
PostgreSQL:*

*Graphical user interface, application

Description automatically generated*

*Дасгал 2. Дараах үр дүнг гаргах асуулга бич. (зөвхөн нэг DBMS дээр байж болно)*

*1.StaffInfo хүснэгт үүсгэ. staffNo, fullName, position, branchAddress, propCount, clientCount гэсэн багануудтай. Хүснэгтэнд query ашиглан өгөгдөл оруул.*

*Graphical user interface, text

Description automatically generated*

*2. Үл хөдлөх хөрөнгийн үнийг 25%-аар нэмэгдүүл.*

*Table

Description automatically generated*

*3. B003 салбарын хариуцан ажиллаж байгаа үл хөдлөх хөрөнгийн үнийг 15%-аар нэмэгдүүл.*

*Table

Description automatically generated with medium confidence*

*4. Tina Murphy-н үл хөдлөх хөрөнгийн үнийг 410, өрөөний тоог 4 болго.*

*Graphical user interface, text, application

Description automatically generated*

*5. Үл хөдлөх хөрөнгийн нөөц хүснэгтээс хариуцах ажилтангүй мэдээллийг арилга.*

*Graphical user interface, application

Description automatically generated*

*6. Үл хөдлөх хөрөнгийн нөөц хүснэгтээс Carol-ийн үл хөдлөх хөрөнгийн мэдээллийг арилга.*

*Graphical user interface, application

Description automatically generated*

*7. Үл хөдлөх хөрөнгийн нөөц хүснэгтээс B003 салбарын 4 өрөөтэй орон сууцны мэдээллийг устга*

*Graphical user interface, application

Description automatically generated*

*Дасгал 3.  
a. Database Systems A Practical Approach to Design, Implementation, and Management номны дасгал 5.23 –д харуулсан Номын сангийн өгөгдлийн санг үүсгэх асуулга бич /DDL/*

*CREATE DATABASE Library;*

*CREATE TABLE Book (*

*ISBN VARCHAR(15) NOT NULL,*

*title VARCHAR(15) NOT NULL,*

*edition VARCHAR(15),*

*year DATE,*

*CONSTRAINT Book\_PK*

*PRIMARY KEY (ISBN)*

*);*

*CREATE TABLE BookCopy (*

*copyNo VARCHAR(15) NOT NULL,*

*ISBN VARCHAR(15) NOT NULL,*

*available VARCHAR(15),*

*CONSTRAINT BookCopy\_PK*

*PRIMARY KEY (copyNo),*

*CONSTRAINT BookCopy\_Book\_FK*

*FOREIGN KEY (ISBN)*

*REFERENCES Book(ISBN),*

*);*

*CREATE TABLE Borrower (*

*borrowerNo VARCHAR(15) NOT NULL,*

*borrowerName VARCHAR(15) NOT NULL,*

*borrowerAddress VARCHAR(15),*

*CONSTRAINT Borrower\_PK*

*PRIMARY KEY (borrowerNo)*

*);*

*CREATE TABLE BookLoan (*

*copyNo VARCHAR(15) NOT NULL,*

*dateOut DATE NOT NULL,*

*dateDue DATE,*

*borrowerNo VARCHAR(15),*

*CONSTRAINT BookLoan\_PK*

*PRIMARY KEY (copyNo,dateOut),*

*CONSTRAINT BookLoan\_Borrower\_FK*

*FOREIGN KEY (borrowerNo)*

*REFERENCES Borrower(borrowerNo),*

*);*

*Graphical user interface, text, application

Description automatically generated*

*b. Database Systems A Practical Approach to Design, Implementation, and Management номны 166 хуудсанд харуулсан Номын сангийн өгөгдлийн санг үүсгэх асуулга бич /DDL/ (Доор оруулав)*

*CREATE DATABASE Hotel;*

*USE Hotel;*

*CREATE TABLE hotel (*

*hotelNo VARCHAR(15) NOT NULL,*

*hotelName VARCHAR(15),*

*city VARCHAR(15),*

*CONSTRAINT hotel\_PK*

*PRIMARY KEY (hotelNo)*

*);*

*CREATE TABLE Room (*

*roomNo VARCHAR(15) NOT NULL,*

*hotelNo VARCHAR(15),*

*type VARCHAR(15),*

*price DECIMAL(9,2),*

*CONSTRAINT Room\_PK*

*PRIMARY KEY (roomNo,hotelNo),*

*CONSTRAINT Room\_hotel\_FK*

*FOREIGN KEY (hotelNo)*

*REFERENCES hotel(hotelNo),*

*);*

*CREATE TABLE Booking (*

*hotelNo VARCHAR(15) NOT NULL,*

*guestNo VARCHAR(15),*

*dateFrom DATE,*

*dateTo DATE,*

*roomNo SMALLINT,*

*CONSTRAINT Booking\_PK*

*PRIMARY KEY (hotelNo,guestNo,dateFrom),*

*CONSTRAINT Booking\_hotel\_FK*

*FOREIGN KEY (hotelNo)*

*REFERENCES hotel(hotelNo),*

*);*

*CREATE TABLE Guest (*

*guestNo VARCHAR(15) NOT NULL,*

*guestName VARCHAR(15),*

*guestAddress VARCHAR(50),*

*CONSTRAINT Guest\_PK*

*PRIMARY KEY (guestNo)*

*);*

*insert into hotel values('fb01', 'Grosvenor', 'London');*

*insert into hotel values('fb02', 'Watergate', 'Paris');*

*insert into hotel values('ch01', 'Omni Shoreham', 'London');*

*insert into hotel values('ch02', 'Phoenix Park', 'London');*

*insert into hotel values('dc01', 'Latham', 'Berlin');*

*insert into room values(501, 'fb01', 'single', 19);*

*insert into room values(601, 'fb01', 'double', 29);*

*insert into room values(701, 'fb01', 'family', 39);*

*insert into room values(1001, 'fb02', 'single', 58);*

*insert into room values(1101, 'fb02', 'double', 86);*

*insert into room values(1001, 'ch01', 'single', 29.99);*

*insert into room values(1101, 'ch01', 'family', 59.99);*

*insert into room values(701, 'ch02', 'single', 10);*

*insert into room values(801, 'ch02', 'double', 15);*

*insert into room values(901, 'dc01', 'single', 18);*

*insert into room values(1001, 'dc01', 'double', 30);*

*insert into room values(1101, 'dc01', 'family', 35);*

*insert into booking values('fb01', 10001, '04-01-04', '04-08-04', 501);*

*insert into booking values('fb01', 10004, '04-15-04', '05-15-04', 601);*

*insert into booking values('fb01', 10005, '05-02-04', '05-07-04', 501);*

*insert into booking values('fb01', 10002, '05-04-16', '05-29-04', 601);*

*insert into booking values('fb01', 10001, '05-01-04', null, 701);*

*insert into booking values('fb02', 10003, '04-05-04', '04-04-10', 1001);*

*insert into booking values('fb02', 10005, '05-12-04', '05-04-30', 1101);*

*insert into booking values('ch01', 10006, '04-21-04', null, 1101);*

*insert into booking values('ch02', 10002, '04-25-04', '05-06-04', 801);*

*insert into booking values('dc01', 10007, '05-13-04', '05-15-04', 1001);*

*insert into booking values('dc01', 10003, '05-20-04', null, 1001);*

*insert into guest values(10001, 'John Kay', '56 High St, London');*

*insert into guest values(10002, 'Mike Ritchie', '18 Tain St, London');*

*insert into guest values(10003, 'Mary Tregear', '5 Tarbot Rd, Aberdeen');*

*insert into guest values(10004, 'Joe Keogh', '2 Fergus Dr, Aberdeen');*

*insert into guest values(10005, 'Carol Farrel', '6 Achray St, Glasgow');*

*insert into guest values(10006, 'Tina Murphy', '63 Well St, Glasgow');*

*insert into guest values(10007, 'Tony Shaw', '12 Park Pl, Glasgow');*

*Graphical user interface, application

Description automatically generated*

*Дасгал 4. Database Systems A Practical Approach to Design, Implementation, and Management номны дасгал 6.7-6.28 хийж гүйцэтгэх (exercise 6.7-6.28 , page 230-231).*

*6.7 List full details of all hotels.*

*SELECT\*FROM Hotel;*

*Table

Description automatically generated*

*6.8 List full details of all hotels in London.*

*SELECT\*FROM hotel*

*WHERE city = 'London'*

*Graphical user interface, application

Description automatically generated*

*6.9 List the names and addresses of all guests living in London, alphabetically ordered by name*

*SELECT guestname, guestaddress FROM guest*

*WHERE guestaddress LIKE '%London%'*

*ORDER BY guestname;*

*Graphical user interface, text, application

Description automatically generated with medium confidence*

*6.10 List all double or family rooms with a price below £40.00 per night, in ascending order of price.*

*SELECT \* FROM room*

*WHERE price < 40 AND type IN ('Double', 'Family')*

*ORDER BY price;*

*Table

Description automatically generated*

*6.11 List the bookings for which no dateTo has been specified.*

*SELECT \* FROM Booking*

*WHERE dateTo IS NULL;*

*Table

Description automatically generated*

*6.12 How many hotels are there?*

*SELECT COUNT(\*) as count FROM hotel;*

*Graphical user interface, text

Description automatically generated with medium confidence*

*6.13 What is the average price of a room?*

*SELECT COUNT(\*)as count, AVG(price) as avg FROM room;*

*Graphical user interface, text

Description automatically generated with medium confidence*

*6.14 What is the total revenue per night from all double rooms?*

*SELECT SUM(price) as revenue FROM room*

*WHERE type = 'Double';*

*Graphical user interface, text, application

Description automatically generated*

*6.15 How many different guests have made bookings for August?*

*SELECT COUNT(DISTINCT guestno) as count*

*FROM booking*

*WHERE datefrom >= '2004-08-01' AND datefrom <= '2004-08-31';*

*Graphical user interface, application

Description automatically generated*

*6.16 List the price and type of all rooms at the Grosvenor Hotel.*

*SELECT price, type FROM Room*

*WHERE hotelNo = (SELECT hotelNo FROM hotel*

*WHERE hotelName = 'Grosvenor Hotel')*

*Graphical user interface, text, application

Description automatically generated*

*6.17 List all guests currently staying at the Grosvenor Hotel*

*SELECT \* FROM Guest*

*WHERE guestNo IN (*

*SELECT guestNo FROM Booking*

*WHERE hotelNo = (SELECT hotelNo FROM hotel*

*WHERE hotelName = 'Grosvenor'))*

*Text

Description automatically generated with medium confidence*

*6.18 List the details of all rooms at the Grosvenor Hotel, including the name of the guest staying in the room, if the room is occupied.*

*SELECT \* FROM Room*

*WHERE hotelNo = (SELECT hotelNo FROM hotel*

*WHERE hotelName = 'Grosvenor')*

*Table

Description automatically generated*

*6.19 What is the total income from bookings for the Grosvenor Hotel today?*

*SELECT SUM(price) as price FROM room WHERE roomNo = (SELECT roomNo FROM Booking WHERE hotelNo = (SELECT hotelNo*

*FROM Hotel WHERE hotelName = 'Grosvenor') AND dateTo is NULL) and hotelNo = (SELECT hotelNo*

*FROM Hotel WHERE hotelName = 'Grosvenor')  
Graphical user interface, text, application, email

Description automatically generated*

*6.20 List the rooms that are currently unoccupied at the Grosvenor Hotel.*

*SELECT r.\* FROM Room r, Booking b*

*WHERE r.roomNo = b.roomNo and r.hotelNo = b.hotelNo and b.dateTo is NOT NULL*

*Graphical user interface, text

Description automatically generated with medium confidence*

*6.21 What is the lost income from unoccupied rooms at the Grosvenor Hotel?*

*SELECT SUM(price) as lost FROM Room r, Booking b*

*WHERE r.roomNo = b.roomNo and r.hotelNo = b.hotelNo and b.dateTo is NOT NULL*

*Graphical user interface, text, application

Description automatically generated*

*6.22 List the number of rooms in each hote*

*SELECT hotelno, COUNT(roomno) AS count FROM room*

*GROUP BY hotelno;*

*Graphical user interface, text, application

Description automatically generated*

*6.23 List the number of rooms in each hotel in London.*

*SELECT hotelNo, COUNT(roomNo) as count FROM Room*

*WHERE hotelNo IN (SELECT hotelNo FROM Hotel WHERE city = 'London')*

*GROUP BY hotelNo*

*Graphical user interface, text, application, email

Description automatically generated*

*6.24 What is the average number of bookings for each hotel in August?*

*SELECT r.hotelNo, r.roomNo, price\*30 as price FROM Room r, Booking b*

*WHERE r.roomNo = b.roomNo and r.hotelNo = b.hotelNo and b.dateFrom > '2004-08-01'and b.dateTo >'2004-08-30'*

*Text

Description automatically generated*

*6.25 What is the most commonly booked room type for each hotel in London?*

*SELECT MAX(count1) AS most*

*FROM( SELECT b.hotelNo, b.roomNo, COUNT(b.roomNo) as count1 FROM Booking b, room r*

*WHERE b.hotelNo = r.hotelNo and b.roomNo = r.roomNo and*

*r.hotelNo IN(SELECT hotelNo FROM Hotel WHERE city = 'London')*

*GROUP BY b.hotelNo, b.roomNo) AS Dummy;*

*Graphical user interface, application

Description automatically generated*

*6.26 What is the lost income from unoccupied rooms at each hotel today?*

*SELECT b.hotelNo, SUM(r.price) as price FROM Booking b, room r*

*WHERE b.hotelNo = r.hotelNo and b.roomNo = r.roomNo and dateTo is not NULL*

*GROUP BY b.hotelNo*

*Graphical user interface, text, application, email

Description automatically generated*

*6.27 Insert rows into each of these tables.*

*Text

Description automatically generated*

*6.28 Update the price of all rooms by 5%*

*SELECT\*FROM ROOM*

*UPDATE room SET price = price\*1.05;*

*SELECT\*FROM ROOM*

*Table

Description automatically generated*

*Даалгавар 5*

*6.41 List all book titles.*

*SELECT title*

*FROM Book;*

*6.42 List all borrower details.*

*SELECT \**

*FROM Borrower;*

*6.43 List all books titles published between 2010 and 2014.*

*SELECT \**

*FROM bookWHERE year BETWEEN ‘2010’ AND ‘2014’;*

*6.44 Remove all books published before 1950 from the database.*

*DELETE*

*FROM bookWHERE year < ‘1950’;*

*6.45 List all book titles that have never been borrowed by any borrower.*

*SELECT \**

*FROM BOOKWHERE ISBN NOT IN*

*(SELECT ISBNFROM BOOKCOPY C, BOOKLOAN LWHERE L.copyno=C.copyno);*

*6.46 List all book titles that contain the word ‘database’ and are available for loan.*

*WHERE b.ISBN=c.ISBN AND b.title like ‘%database%’ AND*

*c.available=’yes’;*

*6.47 List the names of borrowers with overdue books.*

*SELECT borrowerName*

*From Borrower bw, BookLoan bl*

*WHERE bw.borrowerNo = bl.borrowerNoand dateDue> today’s date;*

*6.48 How many copies of each book title are there?*

*SELECT b.title, count (b.copyno) AS numberOfcopies*

*FROM book b, bookcopy c*

*WHERE b.ISBN=c.ISBNGroup by b.title;*

*6.49 How many copies of ISBN “0-321-52306-7” are currently available?*

*SELECT COUNT(copyNo)*

*FROM BookCopy*

*WHERE available = 'Y' AND ISBN = '0-321-52306-7';*

*6.50 How many times has the book title with ISBN “0-321-52306-7” been borrowed?*

*SELECT COUNT(\*)*

*FROM BookCopy bc, BookLoan bl*

*WHERE bc.copyNo = bl.copyNo AND ISBN = '0-321-52306-7';*

*6.51 Produce a report of book titles that have been borrowed by “Peter Bloomfield.”*

*SELECT DISTINCT title*

*FROM Borrower bw, Book b, BookCopy bc, BookLoan bl*

*WHERE bw.borrowNo = bl.borrowerNo AND bl.copyNo = bc.copyNo*

*AND bc.ISBN = b.ISBN AND borrowerName = 'Peter Bloomfield';*

*6.52 For each book title with more than three copies, list the names of library members who have borrowed them.*

*Text

Description automatically generated with low confidence*

*6.53 Produce a report with the details of borrowers who currently have books overdue.*

*SELECT borrowerName, borrowerAddress*

*FROM Borrower bw, BookLoan bl*

*WHERE bw.borrowerNo = bl.borrowerNo AND dateDue > today's date;*

*6.54 Produce a report detailing how many times each book title has been borrowed.*

*SELECT ISBN, COUNT(\*)*

*FROM BookCopy bc, BookLoan bl*

*WHERE bc.copyNo = bl.copyNo*

*GROUP BY ISBN;*