**• SQL statements for schema creation**

AUTHORS

CREATE TABLE AUTHORS

(

ANAME VARCHAR(22) NOT NULL,

ANO VARCHAR(28) NOT NULL ,

CONSTRAINT AUTHORS\_PK PRIMARY KEY(ANO)

);

CREATE TABLE BOOKS

(

BTITLE VARCHAR(28) NOT NULL ,

BID VARCHAR(6) NOT NULL,

CONSTRAINT BOOKS\_PK PRIMARY KEY(BID)

);

CREATE TABLE EARNING

(

ANO VARCHAR(8) NOT NULL,

BID VARCHAR(6) NOT NULL,

Payment INTEGER,

CONSTRAINT EARNING\_FK FOREIGN KEY (ANO) REFERENCES AUTHORS(ANO),

CONSTRAINT EARNING\_FK\_D FOREIGN KEY (BID) REFERENCES BOOKS(BID)

);

**• SQL statements for data insertions**

insert all

into authors (ANAME,ANO)values ('L Grassy','A1')

into authors (ANAME,ANO)values ('D Flower','A2')

into authors (ANAME,ANO)values ('M Trunk', 'A3')

into authors (ANAME,ANO)values ('K Rooty', 'A4')

into authors (ANAME,ANO)values ('T Branch', 'A5')

into authors (ANAME,ANO)values ('L Leafy', 'A6')

into authors (ANAME,ANO)values ('X Bark', 'A7')

select \* from dual;

insert all

into books (BTITLE,BID)values ('Long walk','B1')

into books (BTITLE,BID)values ('Longest day','B2')

into books (BTITLE,BID)values ('Love the fun', 'B3')

into books (BTITLE,BID)values ('Laugh and cry', 'B4')

into books (BTITLE,BID)values ('Laugh all the way', 'B5')

select \* from dual;

insert all

into EARNING (ANO,BID,Payment)values ('A1','B1',78)

into EARNING (ANO,BID,Payment)values ('A1','B2',80)

into EARNING (ANO,BID,Payment)values ('A2','B2',83)

into EARNING (ANO,BID,Payment)values ('A2','B3',88)

into EARNING (ANO,BID,Payment)values ('A2','B4',86)

into EARNING (ANO,BID,Payment)values ('A3','B2',74)

into EARNING (ANO,BID,Payment)values ('A3','B3',89)

into EARNING (ANO,BID,Payment)values ('A4','B2',75)

into EARNING (ANO,BID,Payment)values ('A4','B3',85)

into EARNING (ANO,BID,Payment)values ('A4','B4',55)

into EARNING (ANO,BID,Payment)values ('A5','B1',58)

into EARNING (ANO,BID,Payment)values ('A5','B3',56)

into EARNING (ANO,BID,Payment)values ('A5','B4',55)

into EARNING (ANO,BID,Payment)values ('A6','B2',77)

into EARNING (ANO,BID,Payment)values ('A6','B3',88)

into EARNING (ANO,BID,Payment)values ('A6','B4',54)

select \* from dual;

**• SQL queries and their answers**

**Q1 Find the names of authors who did not co-author**

any book that has ‘K Rooty’ as a co-author.

select e1.bid,a1.aname from authors a1 join earning e1 on e1.ano=a1.ano where e1.bid not in

(select e.bid from authors a join earning e on a.ano=e.ano

where a.ano =(select ano from authors where aname='K Rooty'));

|  |  |
| --- | --- |
| BID | ANAME |
| B1 | L Grassy |
| B1 | T Branch |

**Q2 Find the names of books having both ‘K Rooty’**

and `T Branch' as co-authors.

select btitle from books where bid in

(select e.bid from authors a join earning e on a.ano=e.ano

where a.ano in (select ano from authors where aname='K Rooty')

INTERSECT

select e.bid from authors a join earning e on a.ano=e.ano

where a.ano in (select ano from authors where aname='T Branch'));

|  |
| --- |
| BTITLE |
| Love the fun |
| Laugh and cry |

**Q3 Find the names of authors who co-authored all of**

the books that have ‘K Rooty’ as a co-author.

select a1.aname from authors a1 join earning e1 on a1.ano=e1.ano group by a1.aname

having count(distinct e1.bid)=(select count(distinct e.bid) from authors a join earning e on a.ano=e.ano

where a.ano =(select ano from authors where aname='K Rooty')) and a1.aname!='K Rooty';

|  |
| --- |
| ANAME |
| D Flower |
| T Branch |
| L Leafy |

**Q4 Find the names of authors who earned more**

**money than ‘K Rooty’ in every book that has ‘K Rooty’**

**as a co-author.**

select e1.ano,e1.bid,e1.payment from earning e1 JOIN

(select e.ano,e.bid,e.payment from authors a join earning e on a.ano=e.ano

where a.ano in (select ano from authors where aname='K Rooty')) r

on e1.bid=r.bid

where e1.payment>r.payment ;

|  |  |  |
| --- | --- | --- |
| ano | bid | payment |
| A1 | B2 | 80 |
| A2 | B2 | 83 |
| A6 | B2 | 77 |
| A2 | B3 | 88 |
| A3 | B3 | 89 |
| A6 | B3 | 88 |
| A2 | B4 | 86 |

**Q5 For each book, list the book title, BID, the average**

**payment for all co-authors of the book and the total**

**number of co-authors of the book.**

select b.btitle,e.bid,avg(e.payment) as avg\_payment,count(e.ano) as total\_co\_authors from books b join earning e on b.bid=e.bid group by b.btitle,e.bid;

|  |  |  |  |
| --- | --- | --- | --- |
| BTITLE | BID | AVG\_PAYMENT | TOTAL\_CO\_AUTHORS |
| Long walk | B1 | 68 | 2 |
| Longest day | B2 | 77.8 | 5 |
| Love the fun | B3 | 81.2 | 5 |
| Laugh and cry | B4 | 62.5 | 4 |

**Q6 Extend Q5 by also including the number of coauthors whose payment is below the average**

**payment of the co-authors of the book.**

SELECT bid, AVG(payment) AVG\_Payment, COUNT(ANO) AS NUM\_AUTHORS,

(SELECT COUNT(ANO) FROM earning WHERE payment < AVG(e.payment) AND bid = e.bid)AS PAY\_BELOW\_AVG FROM earning as e GROUP BY bid;

**Q7 For each author, list the author’s name and his/her**

**average payment in all books he/she is a co-author**

**(according to the database).**

select a.aname,avg(e.payment) as avg\_payment from authors a join earning e on a.ano=e.ano group by a.aname;

|  |  |
| --- | --- |
| ANAME | AVG\_PAYMENT |
| L Grassy | 79 |
| D Flower | 85.66666667 |
| M Trunk | 81.5 |
| K Rooty | 71.66666667 |

**Q8 Find the names of authors who have the highest**

**average payment over the books they co-authored in.**

**This cannot be done using “Order By".**

SELECT authors.ANAME, AVG(Payment) AS MAX\_AVG\_PAY FROM authors,earning WHERE authors.ANO = earning.ANO GROUP BY authors.ANO, ANAME

HAVING ROUND(AVG(payment), 2) = (SELECT ROUND(MAX(AVG\_Payment),2) FROM (SELECT AVG(payment) AS AVG\_Payment FROM earning GROUP BY ANO) As eu);