MSBD 6000L: Database Systems

Assignment 2: Table Creation and SQL Queries—Grading Scheme and Example Solution

PART 1: TABLE CREATION

The main requirement for this part is to specify all applicable constraints and to create the tables in the correct order so that the referential integrity (foreign key) constraints can be validated by the DBMS.

Grading Scheme for Part 1

Tables	Create order (marks)	Primary key (marks)	Foreign key (number)	Constraints			Total Marks
				Not Null (number)	Default (number)	Check/Unique (number)	
Clubmember	0.5	0.4	1	3	0	1	3.7
Employee	0.5	0.4	0	3	0	2	4.2
Event	0.5	0.4	1	10	4	6	12.3
Fanclub	0.5	0.4	0	3	0	0	1.2
Hosts	0.5	0.8	2	0	0	0	3.3
MemberOf	0.5	0.8	2	2	0	1	5
RegistersFor	0.5	0.8	2	1	1	2	6.5
RegisteredUser	0.5	0.4	0	5	0	4	7.4
Remark	0.5	0.4	2	6	0	2	6.5
						Max marks	50

For create order, *marks are allocated for each table* as indicated in the grading scheme. The create order is: Employee before {Event, Remark}; Fanclub before {Hosts, MemberOf, Remark}; RegisteredUser before {Clubmember, RegistersFor, Remark}; Clubmember before {MemberOf, Remark}; Event before {Hosts, RegistersFor, Remark}.

For the primary key mark, the grading scheme lists *the marks for the primary key* in each table.

For the foreign key mark, the grading scheme lists *the number of foreign key constraints* in each table. Each foreign key constraint is worth 1 mark.

For the constraints mark, the grading scheme lists *the number of constraints* of each type in each table. Each Not Null constraint is worth 0.1 marks, each Default constraint is worth 0.1 marks, and each Check/Unique constraint is worth 1.5 marks.

PART 2: SQL QUERIES

For each query, an example SQL statement and the expected result when executed on the sample database is given. Different SQL statements that obtain the same result are also acceptable solutions. The Assignment2DB.sql script file can be used to test your own queries to see if they get the expected result.

1. Find the fanclub name and the name of the event for those events that are available for registration and for which no one has registered. Order the result first by fanclub name ascending and then by event name ascending. **[6 marks]**

```
select clubName, eventName
from Fanclub natural join Hosts natural join Event
where status='available'
        and eventId not in (select eventId
                            from RegistersFor)
order by clubName, eventName;
```

Expected Result

clubName	eventName
Jay-Z Fan Club	Jay-Z Photo and Autograph Session
Jay-Z Fanclub	National Day Gala Concert
The Beyhive	National Day Gala Concert
The Swifters	National Day Gala Concert

2. Find the username, first name, last name, occupation and education level of clubmembers who are not a member of any fanclub and who also have not registered for any event. Order the result by last name ascending. **[6 marks]**

```
select username, firstName, lastName, occupation, educationLevel
from RegisteredUser natural join Clubmember natural join Education
where username not in (select username
                       from MemberOf)
      and username not in (select username
                           from RegistersFor)
order by lastName;
```

Expected Result

username	firstName	lastName	occupation	educationLevel
zoeymo	Zoey	Mo	professor	post tertiary

3. Find the username, first name, last name, education level and number of fanclubs of which they are a member for the clubmembers who are members of the most fanclubs. Order the result by username ascending. **[10 marks]**

```
select username, firstName, lastName, educationLevel, count(*) as numClubs
from RegisteredUser natural join ClubMember natural join MemberOf
group by username, firstName, lastName, educationLevel
having count(*)=(select max(count(*))
                 from MemberOf
                 group by username)
order by username;
```

Expected Result

username	firstName	lastName	educationLevel	numClubs
lesterlo	Lester	Lo	tertiary	9
susansze	Susan	Sze	secondary	9
xavierxie	Xavier	Xie	secondary	9

4. Find the name of the fanclub, the total number of their clubmembers who have either a tertiary or a post tertiary education level and their total number of clubmembers for the fanclubs that have the highest total number of members with tertiary or post tertiary education levels. Order the result by club name ascending. **[12 marks]**

```
-- For each fanclub, calculate number of clubmembers who have a tertiary or post tertiary education level.
with HigherEdMemberCount as
(select clubId, count(*) as numHigherEdMembers
 from Fanclub natural join MemberOf natural join Clubmember natural join Education
 where educationLevel in ('tertiary', 'post tertiary')
 group by clubId),
-- For each fanclub, calculate the number of clubmembers.
ClubMembershipCount as
(select clubId, count(*) as numClubmembers
 from MemberOf
 group by clubId)
select club name, numHigherEdMembers, numClubmembers
from Fanclub natural join HigherEdMemberCount natural join ClubMembershipCount
where numHigherEdMembers=(select max(numHigherEdMembers)
                          from HigherEdMemberCount)
order by clubName;
```

Expected Result

clubName	numHigherEdMembers	numClubmembers
BTS Fanclub	5	13
Drake Fanclub	5	12
The Beyhive	5	15

5. Find the event name and the fanclub name where all members of the fanclub hosting the event registered for the event. If an event is hosted by more than one fanclub, then if all the members of one of the fanclubs hosting the event registered for the event, then the event name and name of that fanclub should be in the query result. A fanclub should not be in the query result if not all of its members registered for the jointly hosted event. Order the result by event name ascending. **[16 marks]**

```
with MembersRegisterForEvent as  -- calculate how many members of a club registered for each event
(select Event.eventId, Event.eventname, FanClub.clubId, count(distinct ClubMember.userName) as numRegister
 from Event, RegistersFor, ClubMember, MemberOf, Fanclub, Hosts
 where Event.eventId= RegistersFor.eventId
         and RegistersFor.userName=ClubMember.userName
         and ClubMember.userName=MemberOf.userName
         and MemberOf.clubId=Fanclub.clubId
         and FanClub.clubId=Hosts.clubId
         and Hosts.eventId=Event.eventId
 group by Event.eventId, Event.eventName, Fanclub.clubId)
select eventName, clubName  -- find those events where all club members joined
from Fanclub FC, MembersRegisterForEvent
where FC.clubId= MembersRegisterForEvent.clubId
        and MembersRegisterForEvent.numRegister=(select count(*)  -- calculate the number of club members
                                                 from MemberOf
                                                 where FC.clubId=MemberOf.clubId)
order by eventName;
```

Expected Result

eventName	clubName
Harry Potter: A History of Magic	Pottermore
Superman and MCU Movie Night	Superman Fan Club
Swifters New Year's Eve Party	The Swifters