1. **How can you produce a list of the start times for bookings by members named 'David Farrell'?**

SELECT starttime FROM cd.bookings INNER JOIN cd.members ON cd.members.memid=cd.bookings.memid WHERE cd.members.firstname LIKE 'David' AND cd.members.surname LIKE 'Farrell'

1. **How can you produce a list of the start times for bookings for tennis courts, for the date '2012-09-21'? Return a list of start time and facility name pairings, ordered by the time.**

SELECT bk.starttime AS start , fs.name FROM cd. bookings bk

INNER JOIN cd.facilities fs

ON bk.facid= fs.facid

WHERE fs.name LIKE '%Tennis Court%' AND

bk.starttime BETWEEN '2012-09-21 00:00:00' AND '2012-09-21 23:59:59'

1. **How can you output a list of all members who have recommended another member? Ensure that there are no duplicates in the list, and that results are ordered by (surname, firstname).**

SELECT DISTINCT m2.firstname, m2.surname

FROM cd.members m2 INNER JOIN cd.members m1

ON m2.memid=m1.recommendedby

ORDER BY surname, firstname;

1. **How can you output a list of all members, including the individual who recommended them (if any)? Ensure that results are ordered by (surname, firstname).**

SELECT mem.firstname AS memfname, mem.surname AS memsname, rec.firstname AS recfname, rec.surname AS recsname

FROM cd.members mem LEFT JOIN cd.members rec

ON rec.memid=mem.recommendedby

ORDER BY memsname, memfname;

1. **How can you produce a list of all members who have used a tennis court? Include in your output the name of the court, and the name of the member formatted as a single column. Ensure no duplicate data, and order by the member name.**

SELECT DISTINCT mem.firstname||' '||mem.surname AS member, fac.name as facility

FROM cd.members mem INNER JOIN cd.bookings bk

ON mem.memid=bk.memid

INNER JOIN cd.facilities fac

ON bk.facid = fac.facid

WHERE fac.name LIKE 'Tennis Court%'

ORDER BY member;

1. How can you produce a list of bookings on the day of 2012-09-14 which will cost the member (or guest) more than $30? Remember that guests have different costs to members (the listed costs are per half-hour ‘slot’), and the guest user is always ID 0. Include in your output the name of the facility, the name of the member formatted as a single column, and the cost. Order by descending cost, and do not use any subqueries.

SELECT m.firstname ||' '|| m.surname as member, f.name as facility,

(CASE WHEN m.memid=0 then b.slots\*f.guestcost

else b.slots\*f.membercost END) as cost

FROM cd.members m INNER JOIN cd.bookings b

ON m.memid= b.memid

INNER JOIN cd.facilities f

ON b.facid = f.facid

WHERE b.starttime BETWEEN '2012-09-14 00:00:00' AND '2012-09-14 23:59:59'

AND ( (m.memid = 0 and b.slots\*f.guestcost > 30)

OR (m.memid != 0 and b.slots\*f.membercost > 30))

ORDER BY cost desc;

1. How can you output a list of all members, including the individual who recommended them (if any), without using any joins? Ensure that there are no duplicates in the list, and that each firstname + surname pairing is formatted as a column and ordered.

SELECT DISTINCT firstname|| ' '||surname as member,

(SELECT firstname|| ' '|| surname as recommender

FROM cd.members rec

WHERE rec.memid = mem.recommendedby) FROM cd.members mem

ORDER BY member;

1. The Produce a list of costly bookings exercise contained some messy logic: we had to calculate the booking cost in both the WHERE clause and the CASE statement. Try to simplify this calculation using subqueries. For reference, the question was:

select member, facility, cost from (

select mems.firstname || ' ' || mems.surname as member, facs.name as facility,

case

when mems.memid = 0 then

bks.slots\*facs.guestcost

else

bks.slots\*facs.membercost

end as cost

from

cd.members mems

inner join cd.bookings bks

on mems.memid = bks.memid

inner join cd.facilities facs

on bks.facid = facs.facid

where

bks.starttime >= '2012-09-14' and

bks.starttime < '2012-09-15'

) as bookings

where cost > 30

order by cost desc;

Q2. Ip address: 3.134.112.50

2.7. What is client-server architecture?

Ans. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested, process it and deliver the data packets requested back to the client. One special feature is that the server computer has the potential to manage numerous clients at the same time.

2.8. What is API?

Ans. An application program interface (API) is a set of routines, protocols, and tools for building software applications. Basically, an API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface (GUI) components. A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together.

Q4.

