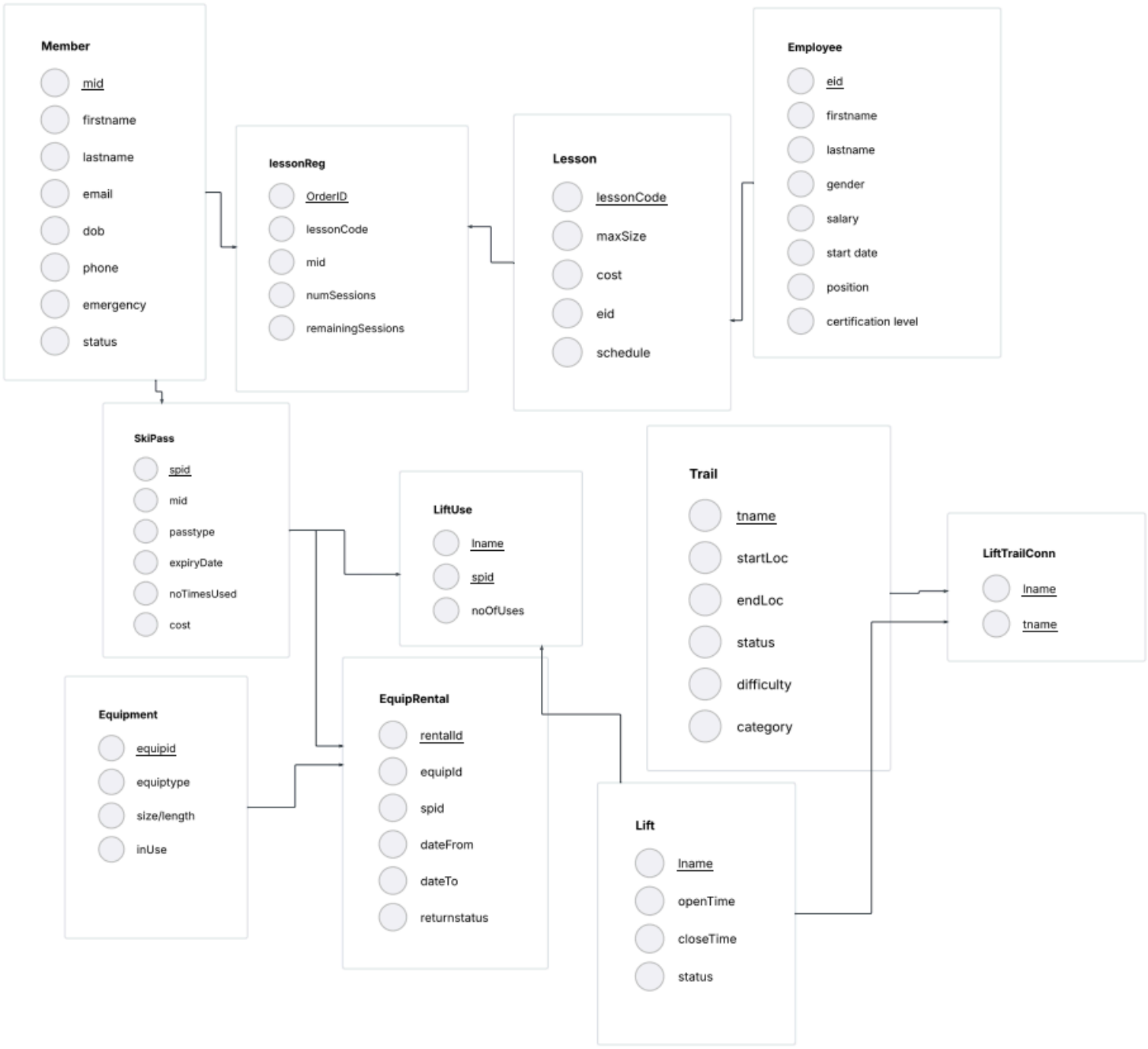
Design

i) Conceptual Database Design



Important notes: The cost of a ski pass is independent of the type because the cost changes based on what time of the year that it is purchased. There is only one group lesson and one private lesson offered by the resort on a given day of the week. We had to change the original ER diagram's equiprental table to have a return status attribute to track whether or not a member had returned their gear.

ii) Logical Database Design

```
SQL> create table bhousmans.lifttrailconn (
  2  lname       varchar2(20),
  3  tname       varchar2(20),
  4  primary key(lname, tname)
  5  );

Table created.

SQL> grant select, insert, update, delete on bhousmans.lifttrailconn to

Grant succeeded.

SQL> create table bhousmans.lift (
  2  lname       varchar2(20),
  3  opentime    integer, --convert int to time ie 715 is 7:15
  4  closetime   integer,
  5  status      integer,
  6  primary key(lname)
  7  );
```

```
SQL> create table bhousmans.trail (
  2  tname       varchar2(20),
  3  startloc    integer,
  4  endloc      integer,
  5  status      integer,
  6  difficulty varchar2(12),
  7  category    varchar2(12),
  8  primary key(tname)
  9  );
```

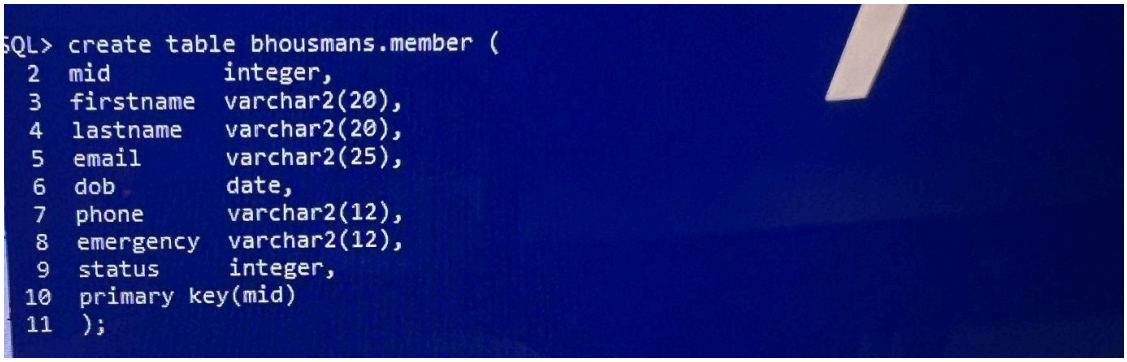
```
SQL> create table bhousmans.employee (
  2  eid         integer,
  3  firstname   varchar2(20),
  4  lastname    varchar2(20),
  5  gender      varchar2(1), --M,F,O
  6  salary      integer,
  7  startdate   date,
  8  position    varchar2(20),
  9  certlevel   integer,
 10  primary key(eid)
 11  );
```

```
SQL> create table bhousmans.equiprental (
  2  rentalid    integer,
  3  equipid     integer,
  4  spid        integer,
  5  datefrom    date,
  6  dateto      date,
  7  primary key(rentalid)
  8  );
```

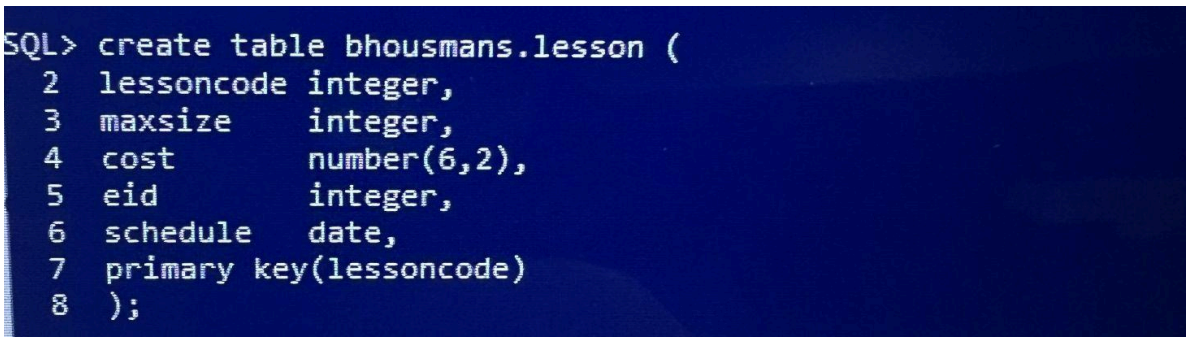
```
SQL> create table bhousmans.lessonreg (
  2  orderid     integer,
  3  lessoncode integer,
  4  mid         integer,
  5  numsessions         integer,
  6  remainingsessions  integer,
  7  primary key(orderid)
  8  );
```

```
SQL> create table bhousmans.equipment (
  2  equipid     integer,
  3  equiptype   varchar2(20),
  4  slength     integer, --size or length. multiply boot size by 10
  5  inuse       integer,
  6  primary key(equipid)
  7  );
```

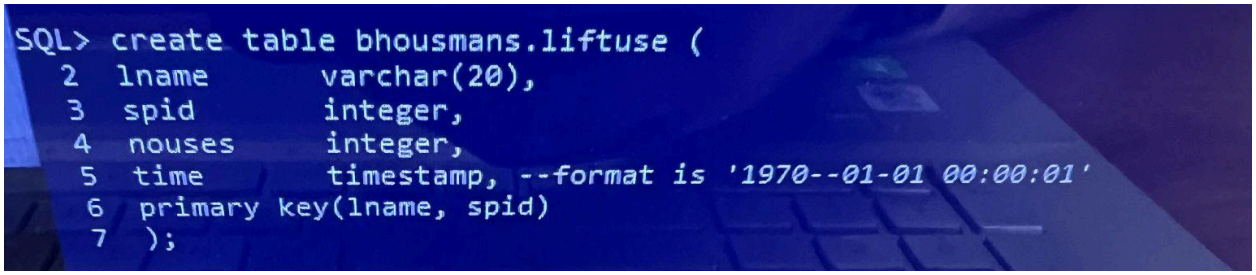
```
SQL> create table bhousmans.skipass (
  2  spid        integer,
  3  mid         integer,
  4  passtype    varchar2(6),
  5  expirydate date,
  6  notimesused         integer,
  7  cost        number(6,2),
  8  primary key(spid)
  9  );
```



```
SQL> create table bhousmans.member (
  2  mid        integer,
  3  firstname  varchar2(20),
  4  lastname   varchar2(20),
  5  email      varchar2(25),
  6  dob        date,
  7  phone      varchar2(12),
  8  emergency  varchar2(12),
  9  status     integer,
 10  primary key(mid)
 11  );
```



```
SQL> create table bhousmans.lesson (
  2  lessoncode integer,
  3  maxsize    integer,
  4  cost       number(6,2),
  5  eid        integer,
  6  schedule   date,
  7  primary key(lessoncode)
  8  );
```



```
SQL> create table bhousmans.liftuse (
  2  lname      varchar(20),
  3  spid       integer,
  4  nouses     integer,
  5  time       timestamp, --format is '1970--01-01 00:00:01'
  6  primary key(lname, spid)
  7  );
```

Note: The equiprental table has an additional attribute called returnstatus that was an integer. This was added after the table was created.

iii) Normalization analysis

We kept the relations in 3NF by making the primary keys of each table the only attribute(s) that functionally determined all of the other attributes for all of the non-trivial FDs.

lifttrailconn table
Attributes: { lname, tname }
FDs-
{ lname, tname } is the primary key and these are the only two attributes, so they are self-determining

Lift table
Attributes: { lname, opentime, closetime, status }
FDs-
lname => opentime

lname => closetime
lname => status

Trail table
Attributes: { tname, startloc, endloc, status, difficulty, category }
FDs-
tname => startloc
tname => endloc
tname => status
tname => difficulty
tname => category

Employee table
Attributes: { eid, firstname, lastname, gender, salary, startdate, position, certlevel }
FDs-
eid => firstname
eid => lastname
eid => gender
eid => salary
eid => startdate
eid => position
eid => certlevel

Equiprental table
Attributes: { rentalid, equipid, spid, datefrom, dateto, returnstatus }
rentalid => equipid
rentalid => spid
rentalid => datefrom
rentalid => dateto
rentailid => returnstatus

Lessonreg table
Attributes: { orderid, lessoncode, mid, numsessions, remainingsessions }
FDs-
orderid => lessoncode
orderid => mid
orderid => numsessions
orderid => remainingsessions

Equipment table
Attributes: { equipid, equiptype, slength, inuse }
FDs-
equipid => equiptype
equipid => slength

equipid => inuse

Skipass table
Attributes: { spid, mid, passtype, expirydate, notimesused, cost }
FDs-
spid => mid
spid => passtype
spid => expirydate
spid => notimesused
spid => cost

Member table
Attributes: { mid, firstname, lastname, email, dob, phone, emergency, status }
FDs-
mid => firstname
mid => lastname
mid => email
mid => dob
mid => phone
mid => emergency
mid => status

Lesson table
Attributes: { lessoncode, maxsize, cost, eid, schedule }
FDs-
lessoncode => maxsize
lessoncode => cost
lessoncode => eid
lessoncode => schedule

Liftuse table
Attributes: { lname, spid, nouses, time }
FDs-
{ lname, spid } => nouses
{ lname, spid } => time

iv) Query Description

The 4th and final query is one that reports all the trails connected by a specified lift, their information, and their statuses to the user. This serves an important role in helping the ski resort visitors know what options are available based on what lift they would like to take. This information is vital in helping the visitors plan their day as they can choose which lifts to take in which order based on the lift timings, trail difficulties, and status of the lifts or trails.

On the technical, database side, this query required one input from the user - the lift name, which was then used in the query to first only the lift required and then it was joined to the lifttrailconn and subsequently trail relations to capture the information of the trails associated with the lifts as well.