DDL:

- create table tablename
  (attribute   datatype   auto-increment   ← if primary key
    attribute   datatype   if not null,

    primary key (attribute),
    foreign key (attribute) reference tablename (attribute));

  * primary key is automatic not null

  * foreign key is also one of attribute

  * each statement separate by comma

  * foreign key need to have <u>not null</u>

- insert into tablename values
  if it has (default , attribute , attribute),
  auto-increment
       (,   ,   ),
       (,   ,   );

- update tablename
  set   statement
  where clause;

- delete from tablename
    where clause   ;

- Referential Integrity

Add FK

Alter table   tablename   ← the change table

add   constraint   FK_tablename   ← the table FK from

foreign key (attribute) references   tablename(attribute)

on delete cascade

on update cascade

Normalization

practice in chatgpt:

1NF

student (studentID , studentName)
     PK

enroll (studentID , courseID   courseName, EnrollmentDate. Grade)
     PFK   PK   PK

2NF

student (studentID , studentName)
     PK

course (courseID , courseName)
     PK

enroll (studentID , courseID , EnrollmentDate , Grade)
     PFK   PFK   PK

3NF

same as above

practice in chatGpt:

1NF

patient (<u>PatientID</u> , patientName)
    PK

record-visit (<u>visitID</u>, DoctorID, DoctorName, Specialization, visitDate
       PK

      <u>TreatmentID</u> , TreatmentName, Cost, <u>PatientID</u>)
         PK   PFK

2NF

patient (<u>PatientID</u> , patientName)
    PK

record-visit (<u>visitID</u>, DoctorID, DoctorName, Specialization,
       PK

      <u>TreatmentID</u> , visitDate, <u>PatientID</u>)
        PFK   PFK
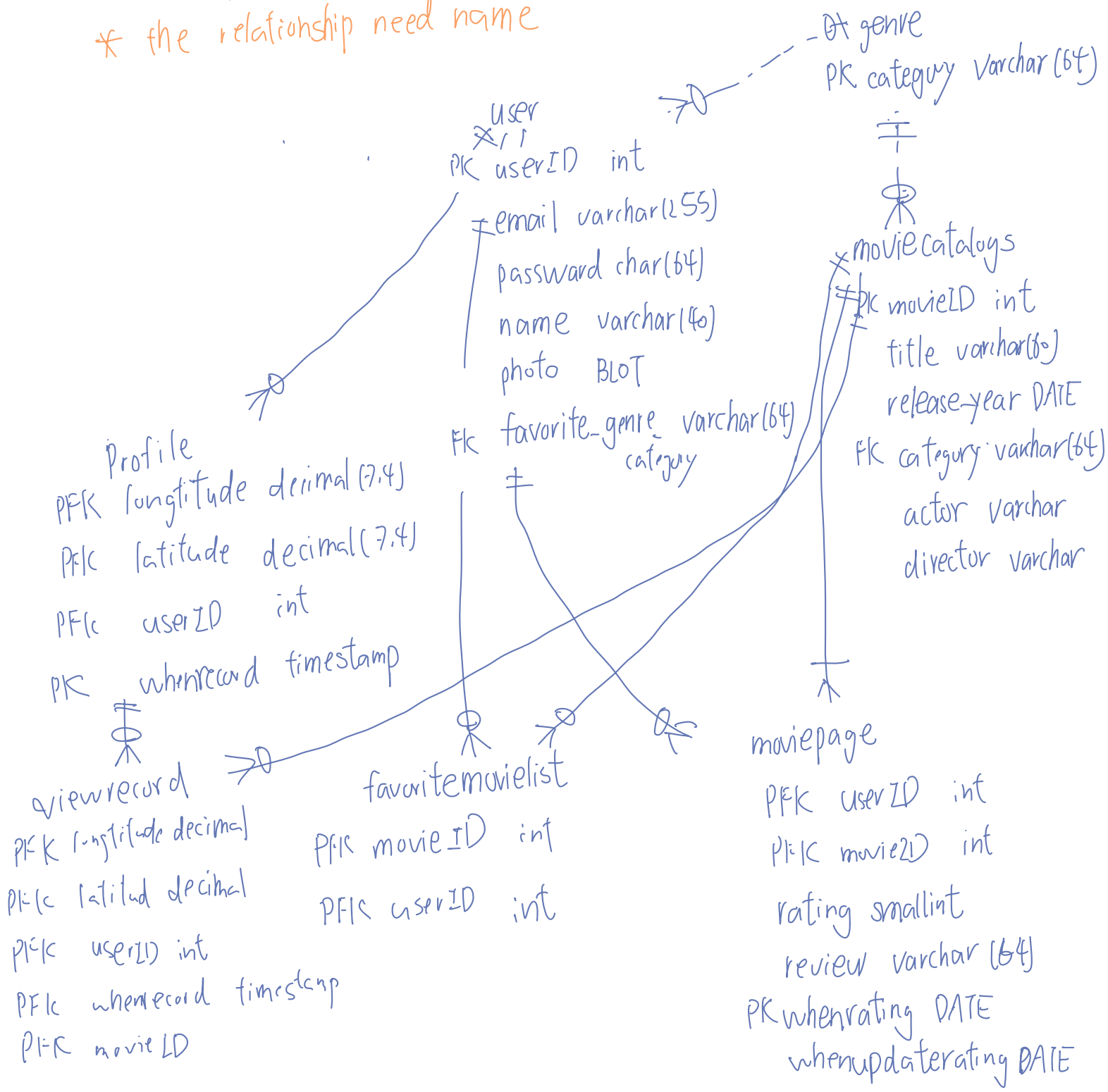
Treatment (<u>TreatmentID</u>, TreatmentName, cost)
         PK

3NF

patient (<u>PatientID</u> , patientName)
    PK

Treatment (<u>TreatmentID</u>, TreatmentName, cost)
         PK

Doctor (<u>DoctorID</u>, DoctorName, Spacialization)
       PK

record~visit (<u>VisitID</u> , DoctorID , <u>TreatmentID</u> , <u>Patient</u>
              PFK   FK   PFK   PFK

                                            visitDate)

ER diagram: In physical model
    * need datatype
    * the relationship need name

- - - - ∘← genre
                PK category varchar (64)

                                        user
                                        PK userID int
                                          email varchar(255)
                                          password char(64)
                                          name varchar(40)
                                          photo BLOT
                                          FK favorite-genre varchar(64)
                                                    category

movie catalogs
PK movieID int
   title varchar(80)
   release-year DATE
   FK category varchar(64)
   actor varchar
   director varchar

Profile
PFK longtitude decimal (7,4)
PFK latitude decimal (7,4)
PFK userID int
PK   whenrecord timestamp

viewrecord
PFK longtitude decimal
PFK latitud decimal
PFK userID int
PFK whenrecord timestamp
PFK movie ID

favoritemovielist
PFK movie ID int
PFK userID int

moviepage
PFK userID int
PFK movieID int
   rating smallint
   review varchar (64)
PK whenrating DATE
   whenupdaterating DATE

DW:

   i. select and explain business process

   ii. declare the grain and justify your choice

   iii. Identify and explain the dimensions.

   iv. identify and explain the fact

⇒ when doing aggregate notice the all PFK need to be the same that the two rows could be aggregate.

⇒ Don't forget to write the data type.