COP 5725 Project Submission 2 – Relational Schema & SQL

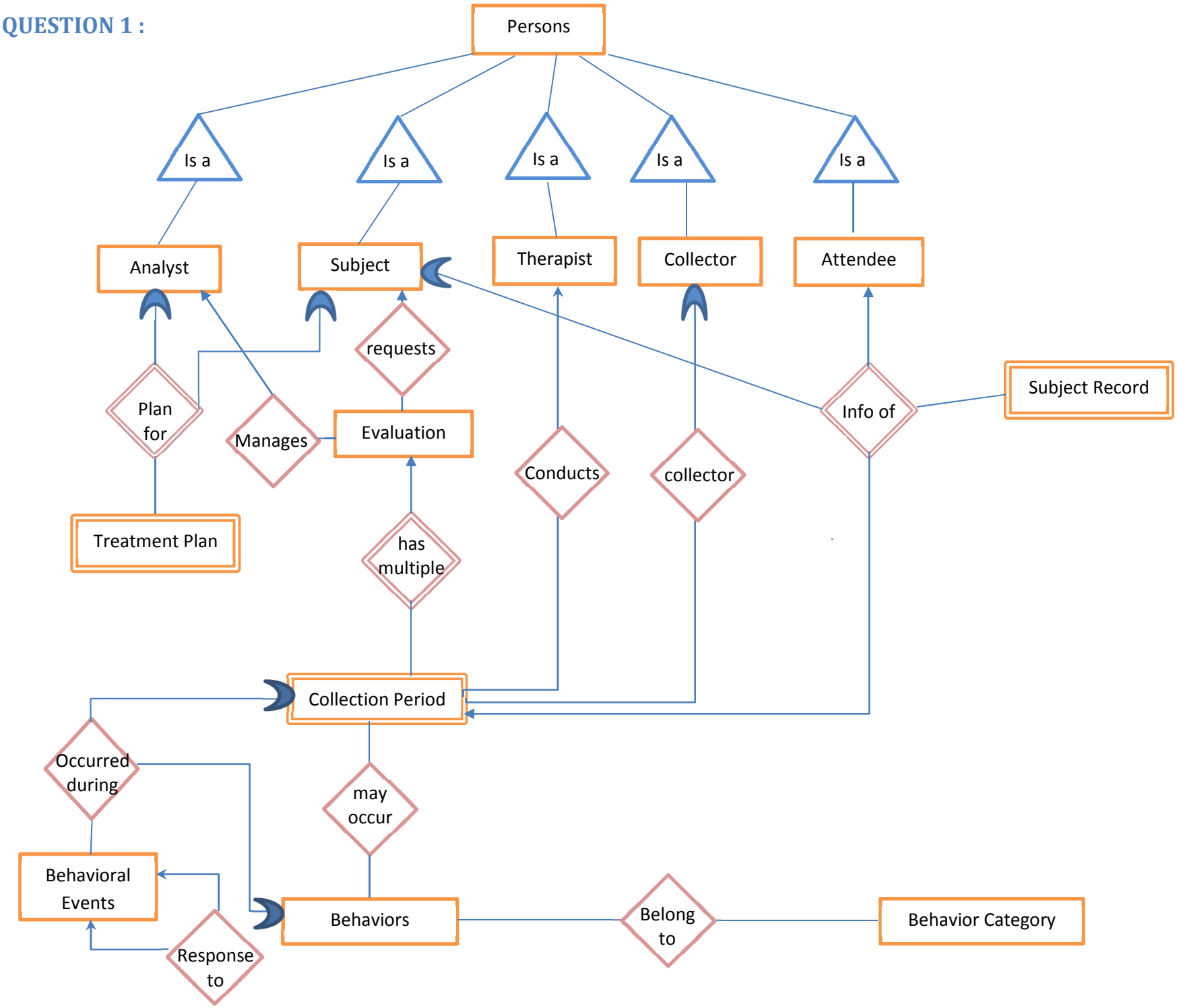
Chandra Shekar Cherukuri
6610-6717

Nitin Gujral
4149-1481

Srirama Tejaswi
7521-6735

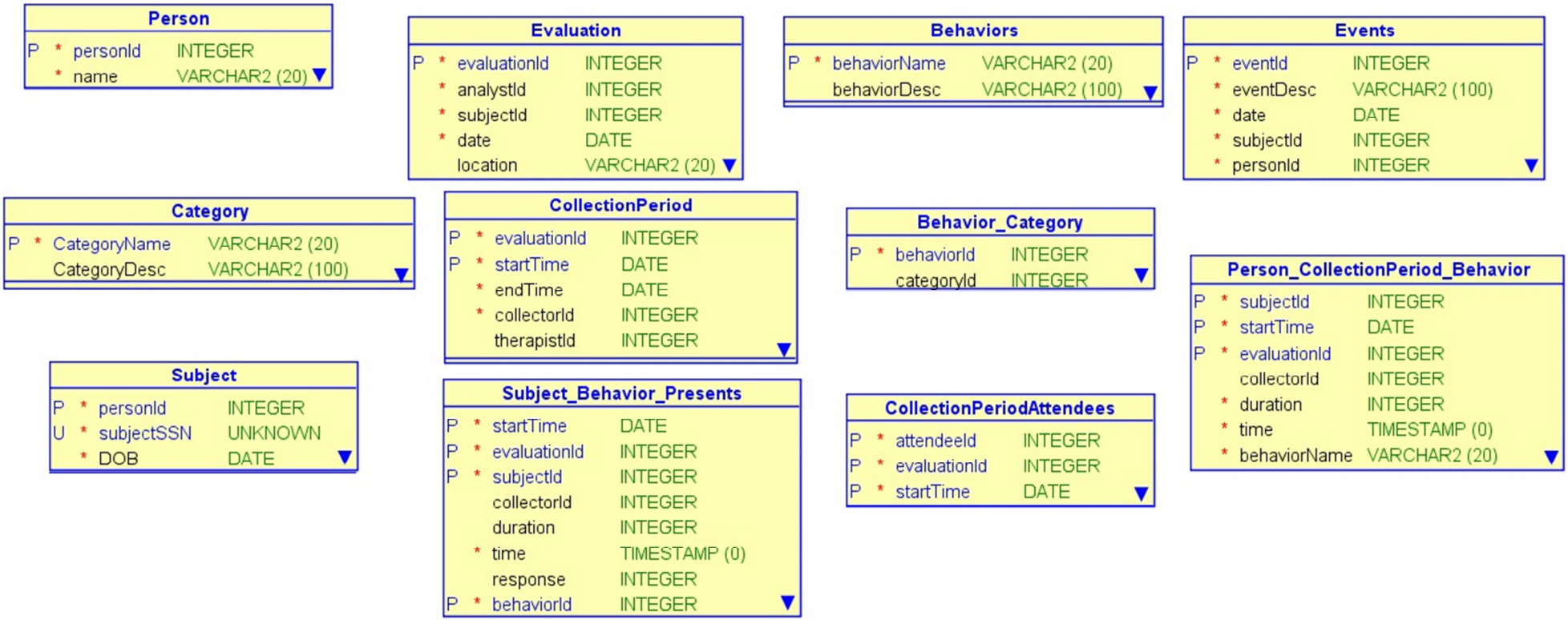
Rajesh Sindhu
4831-2035

QUESTION 1 :

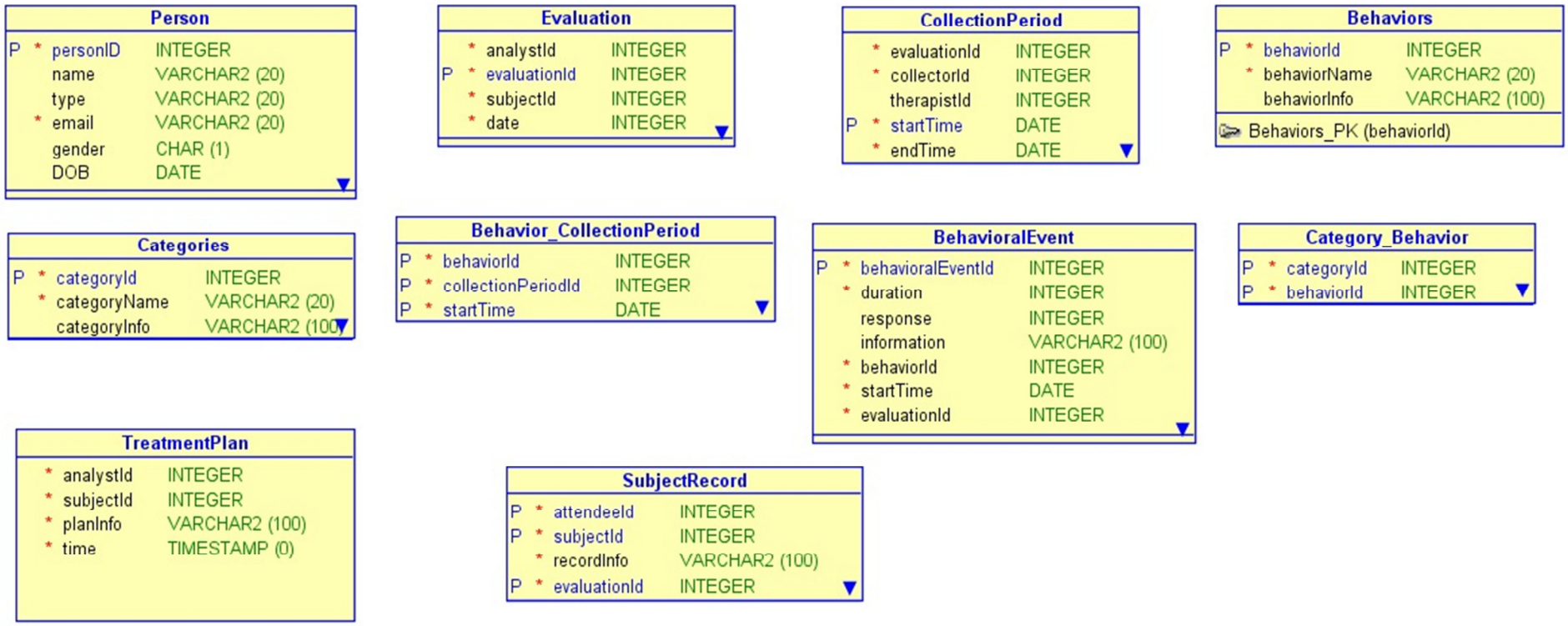

Persons
Is a
Is a
Is a
Is a
Is a
Analyst
Subject
Therapist
Collector
Attendee
requests
Plan for
Manages
Evaluation
Info of
Subject Record
Conducts
collector
Treatment Plan
has multiple
Collection Period
Occurred during
may occur
Behavioral Events
Behaviors
Belong to
Behavior Category
Response to

Entity	Attributes
Evaluations	Evaluation Id
Collection Period	EvaluationId, Start time, end time
Behavior	behaviorId, description
Behavior Categories	categoryId, description
Plan	Plan notes, subjectId
Behavioral Events	Event Id, Occurred-at, response to, behavior Of, Duration
Subject Record	record info
Persons	Name, person Id, type, gender, email, DOB

Relational Schema for attached E-R Diagram



Relational Schema for Our E-R Diagram



QUESTION 4 :

Advantages of our schema:

1. Our ER and Schema are more cleaner and more expressive in terms of entities and what role each entity is performing in the overall schema of things. Entities are more understandable and expressive and entities are very specific in defining the role i.e. there is no ambiguity in design.
2. In the schema presented in item 3, attributes are being duplicated i.e. in the presented schema StudentId, AnalystId, TherapistId and collector id are being used many times. Our schema has eliminated duplication of attributes.
3. In the presented ER, there is no possible way to monitor behaviors within a collection period, which is the major requirement of project. There should be some provision for this; our ER has provision to monitor behaviors within a collection period and generate detailed reports.
4. There is provision for treatment plan in our schema. This will help in maintaining treatment plan for the subject. It will keep records of the entire requirement for treatment of patient.
5. In the ER presented in item 2, since therapist is optional .there will be lots of null attributes in the schema.

Presented schema's advantages:

1. Since there would be lesser joins to extract information from the schema, it would be more efficient during report generation. This is because SubjectId, Therapist Id and AnalystId are being used in most of the tables hence, reducing ambiguity.
2. There is a clear distinction between a subject and other entities because there is separate table for subject in the presented schema.
3. This schema would be efficient for collective analysis, for example: To find all the behavior's portrayed by the subject during a collection period.

QUESTION 5 :

Given Schema Functional Dependencies :

Person table:
PersonId → Name

Subject table:
PersonId → SSN **SSN → DOB,Name**

CollectionPeriod table:
EvaluationId,startTime → TherapistId,endTime,CollecterId **CollecterId, startTime → TherapistId,endTime,EvaluationId**

Evaluation table:
EvaluationId → date,location,AnalystId,SubjectId

Behavior table:
Name → description

Category table:
Name → description

CollectionPeriod_Attendees table:
EvaluationId,startTime → AttendeeId

Event table:
EventId → Description, date-time,subjectId,attendeeId

Person_CollectionPeriod_Behavior table
Personid,startTime,evaluationId,BehaviorName → duration,time

Subject_Behavior_Presents:
SubjectId,startTime,evaluationId,BehaviorName → duration,time,response

Behavior_Category:
No FD

Our Schema Functional Dependencies :

Person:
PersonId → Name,type,email,gender,DOB

Evaluation:
EvaluationId → date, analystId,subjectId

CollectionPeriod:
CollectionPeriodId →EvaluationId,TherapistId,CollectorId,startTime,endTime **EvaluationId,startTime → CollectionPeriodID** **EvaluationId,startTime → TherapistId,CollectorId, endTime**

CollectionPeriod_Evaluation
No FD

Behaviors:
BehaviorId → Behaviorname,Info

Behavior_CollectionPeriod:
No FD

BehavioralEvent:
EventID → CollectionPeriodId,response,duration,info,behaviorId,starttime **CollectionPeriodId,starttime → BehaviorId,response,duration,info**

Categories:
CategoryId → Categoryname,Info

Behavior_Category
No FD

Subject_Record:
AttendeeId,SubjectId,EvaluationId → RecordInfo

TreatmentPlan:
AnalystId,SubjectId →PlanInfo,time

Trivial Dependencies have been omitted.

QUESTION 6 :

Given Schema :

Person table:	
PersonId → Name	**Primary Key**
This table is in BCNF	

Subject table:	
PersonId → SSN, DOB **SSN → DOB**	**Needs To be converted to BCNF.**
This table is not in BCNF because a non-key attribute SSN derives DOB attribute.	
This table is split into : **Person_Subject{PersonId, SSN}** **Subject_Info{SSN, DOB}** **This resolves the Transitive Functional Dependencies.** **This table is now in BCNF**	

CollectionPeriod table:	
EvaluationId,startTime → TherapistId,endTime,CollecterId	**Candidate Key**
This table is in BCNF	

Evaluation table:	
EvaluationId → date,location,AnalystId,SubjectId	**Primary Key**
This table is in BCNF	

Behavior table:	
Name → description	**Primary Key**
This table is in BCNF	

Category table:	
Name → description	**Primary Key**
This table is in BCNF	

CollectionPeriod_Attendees table:	
EvaluationId,startTime → AttendeeId	**Candidate Key**
This table is in BCNF	

Event table:	
EventId → Description, date-time,subjectId,attendeeId	Primary Key
This table is in BCNF	

Person_CollectionPeriod_Behavior table	
Personid,startTime,evaluationId,BehaviorName → duration,time	Candidate Key
This table is in BCNF	

Subject_Behavior_Presents:	
SubjectId,startTime,evaluationId,BehaviorName → duration,	Candidate Key
This table is in BCNF	

Behavior_Category:	
No FD	
This table is in BCNF	

Our Schema :

Person:	
PersonId → Name,type,email,gender,DOB	**Primary Key**
This table is in BCNF	

Evaluation:	
EvaluationId → date, analystId,subjectId	**Primary Key**
This table is in BCNF	

CollectionPeriod:	
EvaluationId, startTime →TherapistId, CollectorId, endTime	**Candidate Key**
This table is in BCNF	

CollectionPeriod_Evaluation	
No FD	
This table is in BCNF	

Behaviors:	
BehaviorId → Behaviorname,Info	**Primary Key**
This table is in BCNF	

Behavior_CollectionPeriod:	
No FD	
This table is in BCNF	

BehavioralEvent:	
EventID → evaluationId, startTime, response, duration, info, behaviorId **EvaluationId, startTime → BehaviorId, response, duration, info**	**Candidate Key**
This table is in BCNF	

Categories:	
CategoryId → Categoryname,Info	Primary Key
This table is in BCNF	

Behavior_Category	
No FD	
This table is in BCNF	

Subject_Record:	
AttendeeId,SubjectId,EvaluationId → RecordInfo	Candidate Key
This table is in BCNF	

TreatmentPlan:	
AnalystId,SubjectId →PlanInfo,time	Candidate Key
This table is in BCNF	

* Trivial Functional Dependencies have been omitted.

QUESTION 7 :

Advantages of our schema after BCNF:

1. Our schema has more clearer attributes and fulfills the specification much better than the BCNF schema derived from the ER diagram given in the question.

2. Schema derived from the ER diagram given in the question has split the behavioral events of Subject and Persons (other than subject). Our schema stored it based on behavioral events rather than basing it on Persons. We believe our schema will be easier to query when we have to select the events of a given collection period.

Presented schema's advantages after BCNF:

1. The maintainence of the tables will be easier as the Schema derived from the ER diagram given in the question has split the behavioral events of Subject and Persons (other than subject).

QUESTION 8 :

SQL Script to generate our schema :

```
CREATE TABLE Person (
     name VARCHAR2 (20)  NOT NULL ,
     personId INTEGER PRIMARY KEY ,
     SSN VARCHAR2 (10)  UNIQUE NOT NULL ,
     type VARCHAR2 (20)  CHECK (type in ('ANALYST', 'THERAPIST', 'SUBJECT',
'COLLECTOR', 'ATTENDEE')) ,
     email VARCHAR2 (20) ,
     gender CHAR (1)  CHECK (gender in ('M','F')) ,
     DOB DATE NOT NULL ,
     check (email like '%@%.%' and SSN like '%-%-%')
    );

CREATE TABLE Evaluation (
     analystId INTEGER  REFERENCES Person(personId),
     evaluationId INTEGER  PRIMARY KEY ,
     subjectId INTEGER  NOT NULL REFERENCES Person(personId),
     "date" DATE  NOT NULL
    );

CREATE TABLE CollectionPeriod
    (
     evaluationId INTEGER  REFERENCES Evaluation(evaluationId) NOT NULL,
     therapistId INTEGER REFERENCES Person(personId),
     collectorId INTEGER  REFERENCES Person(personId) NOT NULL,
     startTime DATE NOT NULL,
     endTime DATE  NOT NULL,
     Constraint CollectionPeriod_PK PRIMARY KEY ( evaluationId, startTime )
    );

CREATE TABLE Behaviors
    (
     behaviorId INTEGER  PRIMARY KEY ,
     behaviorName DATE  NOT NULL ,
     behaviorInfo VARCHAR2 (100)
    );


CREATE TABLE Behavior_CollectionPeriod
    (
     behaviorId INTEGER  REFERENCES Behaviors(behaviorId),
     evaluationId INTEGER,
     collectionPeriodStartTime DATE ,
     CONSTRAINT bcp_fk FOREIGN KEY(evaluationId, collectionPeriodStartTime)
REFERENCES CollectionPeriod(evaluationId, startTime),
     CONSTRAINT bcp_pk PRIMARY KEY(evaluationId, collectionPeriodStartTime)
    ) ;

CREATE TABLE Behavioral_Event
    (
     behavioral_eventId INTEGER  PRIMARY KEY,
```

```
     response INTEGER ,
     duration INTEGER  NOT NULL ,
     information VARCHAR2 (100) ,
     behaviorId INTEGER  REFERENCES Behaviors(behaviorId) NOT NULL,
     time TIMESTAMP (0)  NOT NULL
    );

CREATE TABLE Categories
    (
     categoryId INTEGER  PRIMARY KEY ,
     category_name VARCHAR2 (20)  UNIQUE NOT NULL ,
     category_info VARCHAR2 (100)  NOT NULL
    );

CREATE TABLE Category_Behavior
    (
     categoryId INTEGER  REFERENCES Categories(categoryId) NOT NULL ,
     behaviorId INTEGER  REFERENCES Behaviors(behaviorId) NOT NULL
    ) ;

CREATE TABLE Subject_Record
    (
     attendeeId INTEGER  REFERENCES Person(personId)  NOT NULL ,
     subjectId INTEGER  REFERENCES Person(personId) NOT NULL ,
     recordInfo VARCHAR2 (100)  NOT NULL ,
     evaluationId INTEGER  REFERENCES Evaluation(evaluationId) NOT NULL,
     CONSTRAINT Subject_Record_PK PRIMARY KEY ( attendeeId, subjectId,
evaluationId )
    ) ;

CREATE TABLE TreatmentPlan
    (
     analystId INTEGER  REFERENCES Person(personId) NOT NULL ,
     subjectId INTEGER  REFERENCES Person(personId) NOT NULL ,
     planInfo VARCHAR2 (100)  NOT NULL ,
     time TIMESTAMP (0)  NOT NULL
    ) ;
```