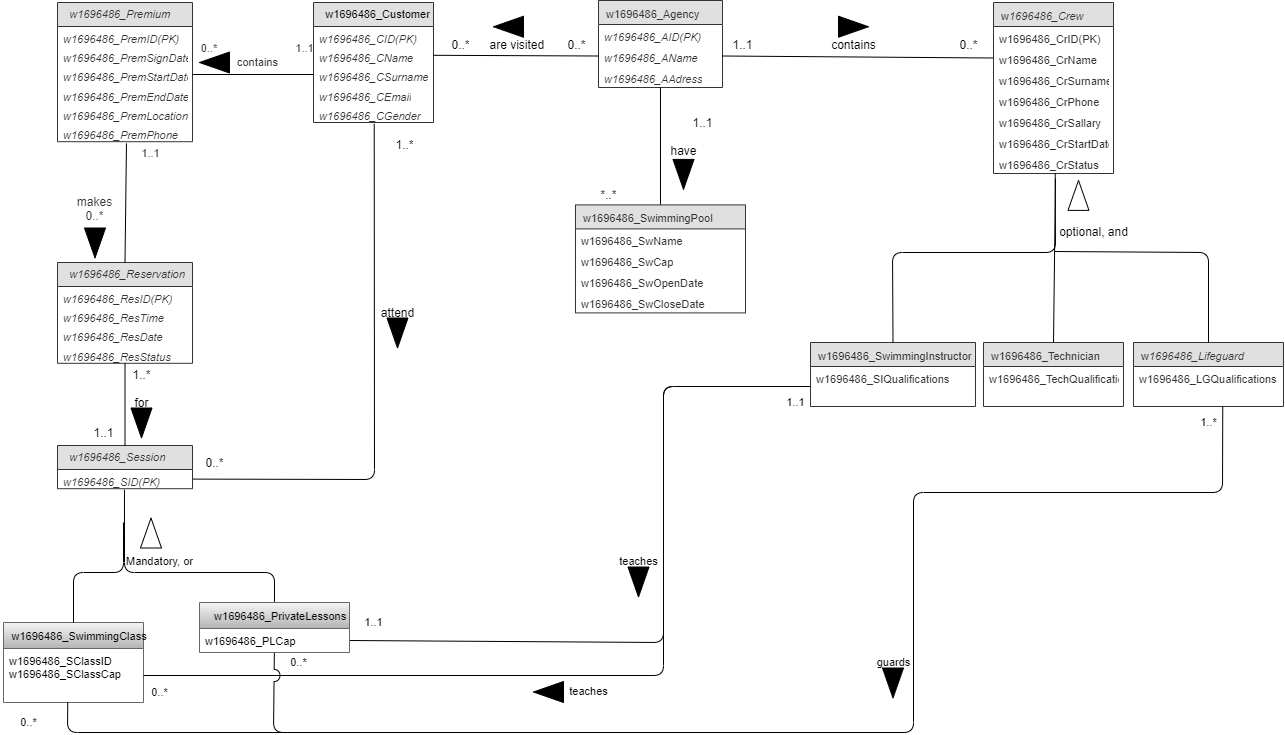
**Part A**



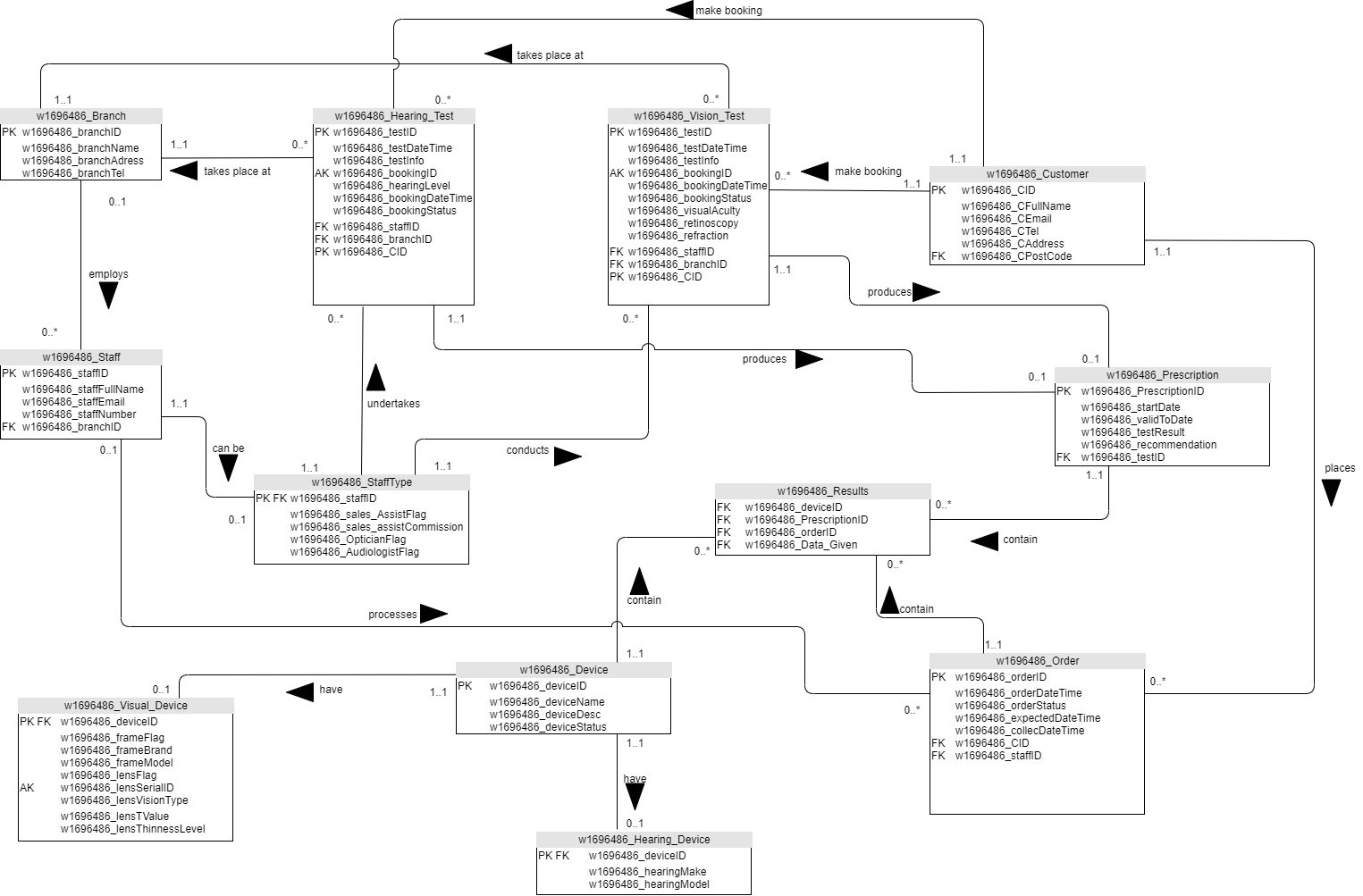
|  |  |
| --- | --- |
| Entity name | Description |
| W1696486\_Agency | A Agency is an organization that gives all that needed for other entities. |
| W1696486\_Customer | Customer are both custumer and premium custumers that bought membership as well as all data like „name..” |
| W1696486\_reservation | reservation are made by premium to attend private lesson session which holds info about date and time and reservation status. |
| W1696486\_Crew | Crew is an Employee of Agency . |
| W1696486\_Session | Session is a holder for premium customer session status. |
| W1696486\_Premium | Premium are unique type of customer who can reserve their session online. |
| W1696486\_SwimmingPool | SwimmingPool is a place for swimming lessons. |

|  |  |  |
| --- | --- | --- |
| General entity | Specialised entity | Explanation |
| W1696486\_Session | W1696486\_SwimmingClass | Groupe lessons with one swimming instructor and many Premium |
| W1696486\_PrivateLessons | private swimming lessons in the swimming pool with the swimming instructor |
| W1696486\_ Crew | W1696486\_SwimmingInstructor | SwimmingInstructor helps premium customer with PrivateLessons session as well as SwimmingClass and teachs them. |
| W1696486\_Lifeguard | Crew that in case of emergency helps customer or premium during sessions. |
| W1696486\_Technician | Technician Crew maintain facilities and takes care of every Agency and makes sure water quality is good. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity name | Multiplicity | Relationship | Multiplicity | Entity name | Justifications for the multiplicity  (4 statements for each relationship) |
| W1696486\_Agency | 0..\* | Are visited | 0..\* | W1696486\_customers | Agency may not be visited by any customer |
| Agency may be visited by many customers |
| Customer may not visit any Agency |
| Customer may visit many Agency |
| W1696486\_ Agency | 1..1 | contains | 0..\* | W1696486\_ Crew | One Agency might have no Crew |
| One Agency might have many Crew |
| One Crew can work at at least one Agency |
| And at most one Agency |
| W1696486\_ Agency | 1..1 | have | \*..\* | W1696486\_swimmingpool | Agency must have many swimmingpool |
| Many to many |
| A swimmingpool can be at most at one a Agency |
| Can be at least at a Agency |
| W1696486\_SwimmingInstructor | 1..1 | teaches | 0..\* | W1696486\_swimmingClass | SwimmingInstructor may not teach any swimmingClasss |
| May teach many swimmingClass |
| A swimmingClass can be taken by at most one SwimmingInstructor |
| And at least one SwimmingInstructor |
| W1696486\_Premium | 1..1 | makes | 0..\* | W1696486\_reservation | Premium may not make any reservation |
| Premium may make many reservations. |
| One reservation for one Premium |
| Can only do own reservation. |
| W1696486\_reservation | 1..\* | for | 1..1 | W1696486\_session | Can reservate for at least one session |
| At most one session |
| For session can make lest one reservation |
| To many reservations |
| W1696486\_Lifeguard | 1..\* | guards | 0..\* | W1696486\_SwimmingClass | Lifeguard may guard no SwimmingClass |
| Lifeguard may guard many SwimmingClass |
| SwimmingClass can be guarded by at least one Lifeguard |
| To at most many SwimmingClass |
| W1696486\_Lifeguard | 1..\* | guards | 0..\* | W1696486\_PrivateLessons | Lifeguard may guard no PrivateLessons |
| Lifeguard may guard many PrivateLessons |
| PrivateLessons can be guarded by at least one Lifeguard |
| To at most many Lifeguards |
| W1696486\_customer | 1..\* | attend | 0..\* | W1696486\_session | Customer may attend no session |
| Customer may attend many session |
| Session can be attended by at least one customer |
| To at most many customers. |
| W1696486\_customer | 1..1 | contains | 0..\* | W1696486\_premium | Customer may not contain any premium |
| May contain many premiums |
| premium can be at least one customer |
| To at most one customer |

|  |  |  |
| --- | --- | --- |
| Entity name | Attributes for this entity (include PK) | Justification |
| W1696486\_Agency | W1696486\_AID{PK} | BID primary key of Agency entity, unique identificator for every record in the table. |
| W1696486\_ Agency | W1696486\_AName | Aname its name of Agancy. |
| W1696486\_ Agency | W1696486\_AAdress | address of Agency. Specific localization of Agancy |
| W1696486\_Customer | W1696486\_CID{PK} | CID is a primary key of Customer entity because it’s unique for each customer and helps to indetify tchem. |
| W1696486\_Customer | W1696486\_CEmail | CEmail is a email address of the customer |
| W1696486\_Customer | W1696486\_CName | CName is the name of customer. |
| W1696486\_Customer | W1696486\_CSurname | CSurname is the Surname of customer. |
| W1696486\_Customer | W1696486\_CGender | Gender of customer |
| W1696486\_premium | W1696486\_PremID{PK} | PremID is a primary key of premium entity, cause it is unique from the other attributes . |
| W1696486\_ premium | W1696486\_PremLocation | PremLocation is the specific address of premium. |
| W1696486\_ premium | W1696486\_PremSignDate | PremSignDate is a date that customer join in the premium status. |
| W1696486\_ premium | W1696486\_PremStartDate | StartDate is the start date for receiving premium services. |
| W1696486\_ premium | W1696486\_PremEndDate | EndDate is the end date for receiving premium services. |
| W1696486\_ Crew | W1696486\_CrID{PK} | CrID is an unique identifier attribute for each employee. |
| W1696486\_ Crew | W1696486\_CrName | CrName is the name of each who work under the Agancy. |
| W1696486\_ Crew | W1696486\_CrPhone | CrPhone is the attribute that keeps phone nr of each employee |
| W1696486\_ Crew | W1696486\_CrAddress | CrAddress keeps records of the Address of the employee |
| W1696486\_ Crew | W1696486\_CrStartDate | CrStartDate keeps records about the join date of the Crews. |
| W1696486\_ Crew | W1696486\_CrSallary | CrSallary is a payment that Crew recives each month. |
| W1696486\_reservation | W1696486\_ResID{PK} | ResID is the primary key of the entity reservation and is a unique identifier key. |
| W1696486\_ reservation | W1696486\_ResDateAndTime | The date and time in which reservation was made. |
| W1696486\_ reservation | W1696486\_ResStatus | Status of the reservation, {pending or confirmed}. |
| W1696486\_Session | W1696486\_SID{PK} | SID {PK} is and unique identifier and the primary key of the entity session. |
| W1696486\_ SwimmingPool | W1696486\_SwName | The name of SwimmingPool |
| W1696486\_ SwimmingPool | W1696486\_SwCap | SwCap is the max number of premium that can be fitted in a swimming pool. |
| W1696486\_ SwimmingPool | W1696486\_SwOpenDate | SwOpenDate is the date that SwimmingPool opens. |
| W1696486\_ SwimmingPool | W1696486\_ SwCloseDate | SwCloseDate is the date that SwimmingPool closes. |
| W1696486\_SwimmingClass | W1696486\_SClassID | SClassID is the number of the swimmingClass. |
| W1696486\_SwimmingClass | W1696486\_ SClassCap | The SClassCap -how many premium can fit. |
| W1696486\_privateLessons | W1696486\_ PLCap | How many premium can take part in each sessions. |
| W1696486\_Lifeguard | W1696486\_ LGQualifications | LGQualifications is an attribute that shows the Qualification level of the Lifeguard. |
| W1696486\_SwimmingInstructor | W1696486\_ SIQualifications | SIQualifications shows the technical qualification of the SwimmingInstructor. |
| W1696486\_Technician | W1696486\_TechQualifications | TechQualifications shows the technical ability of the Technician. |

Part B



**TASK-6**

**Mapping the AudioVizzion Conceptual ERD in to a Logical ERD.**

Converting the Conceptual ERD to a Logical ERD:

1. First, map out all the specialisation that are the most significant ones.
2. Secound, doing the specialisation of {Mandatory, And} which results in a single Entity table combining all the attributes of the two table.
3. Third, the specialisation of {Optional, And} creates two tables and a One to One relationship optional on one side.
4. Fourth, the specialisation of {Mandatory, Or} creates two separate tables where Each table have their own relationships with the rest of the pattern.
5. Fifth, the specialisation of {Optional, Or} creates three tables and two One to One relationships optional on one side.
6. Sixth, complex triple relationship. This is same as many to many but with a triple relationship. Here we get a new linked child table with the three existing parents tables.
7. Seventh, mapping out the tables which has One to One multiplicity mandatory on both sides which results in creating a single table with all the attributes with in it.
8. Eighth, I did the remaining tables having the multiplicity of One to One optional on one side and the One to Many ones where the child table gets a Foreign Key representing the Primary Key of the parent table.

Table named w1696486\_Visual\_Device from the two tables w1696486\_Frame and w1696486\_Lens. And all the attributes from both table w1696486\_Frame and w1696486\_Lens gets in to the table w1696486\_Visual Device. The Alternative key from the table w1696486\_Lens remain same into the new table.

w1696486\_Staff and combining the three sub-entities w1696486\_Sale\_Assistant, w1696486\_Optician and w1696486\_Audiologist we get one child tables named w1696486\_Staff\_Type. The PK for the child table is the same as the parent table. And the FK of the child table references PK of the parent table.

w1696486\_Hearing\_Test and w1696486\_Vision\_Test, where each table contain all the attributes of the super-entities and the sub entities and the PK for both are from the PK of the original Super entity.

w1696486\_Device, with two child table w1696486\_Visual\_Device and w1696486\_Hearing\_Device, where the parent table has One to One relationship optional on one side with the child tables. PK of the child table is the same as the PK for the Parent table and the FK of the Child table reference the PK of the Parent table.

Between the table w1696486\_Customer and w1696486\_Booking, w1696486\_Customer is the strongest entity which will become the parent table and pass it PK as FK to the child table w1696486\_Booking.

Between the table w1696486\_Customer and w1696486\_Order, w1696486\_Customer is the strongest entity which will become the parent table and pass it PK as FK to the child table w1696486\_Order.

Between the table w1696486\_Branch and w1696486\_Staff, w1696486\_Branch is the strongest entity which will become the parent table and pass it PK as FK to the child table w1696486\_Staff.

Between the table w1696486\_Staff and w1696486\_Order, w1696486\_Staff is the strongest entity which will become the parent table and pass it PK as FK to the child table w1696486\_Order.

Between the table w1696486\_Branch and Both w1696486\_Hearing\_Test and w1696486\_Vision\_Test, w1696486\_Branch is the strongest entity which will become the parent table and pass it PK as FK to the child table w1696486\_Hearing\_Test and w1696486\_Vision\_Test.

Between the table w1696486\_Staff\_Type and w1696486\_Hearing\_Test, w1696486\_Staff\_Type is the strongest entity which will become the parent table and pass it PK as FK to the child table w1696486\_Hearing\_Test.

Between the table w1696486\_Staff\_Type and w1696486\_Vision\_Test, w1696486\_Staff\_Type is the strongest entity which will become the parent table and pass it PK as FK to the child table w1696486\_Vision\_Test.

**TASK - 7**

**SQL code for the table w1696486\_Device**

CREATE TABLE w1696486\_Device

(

w1696486\_deviceID INT(3),

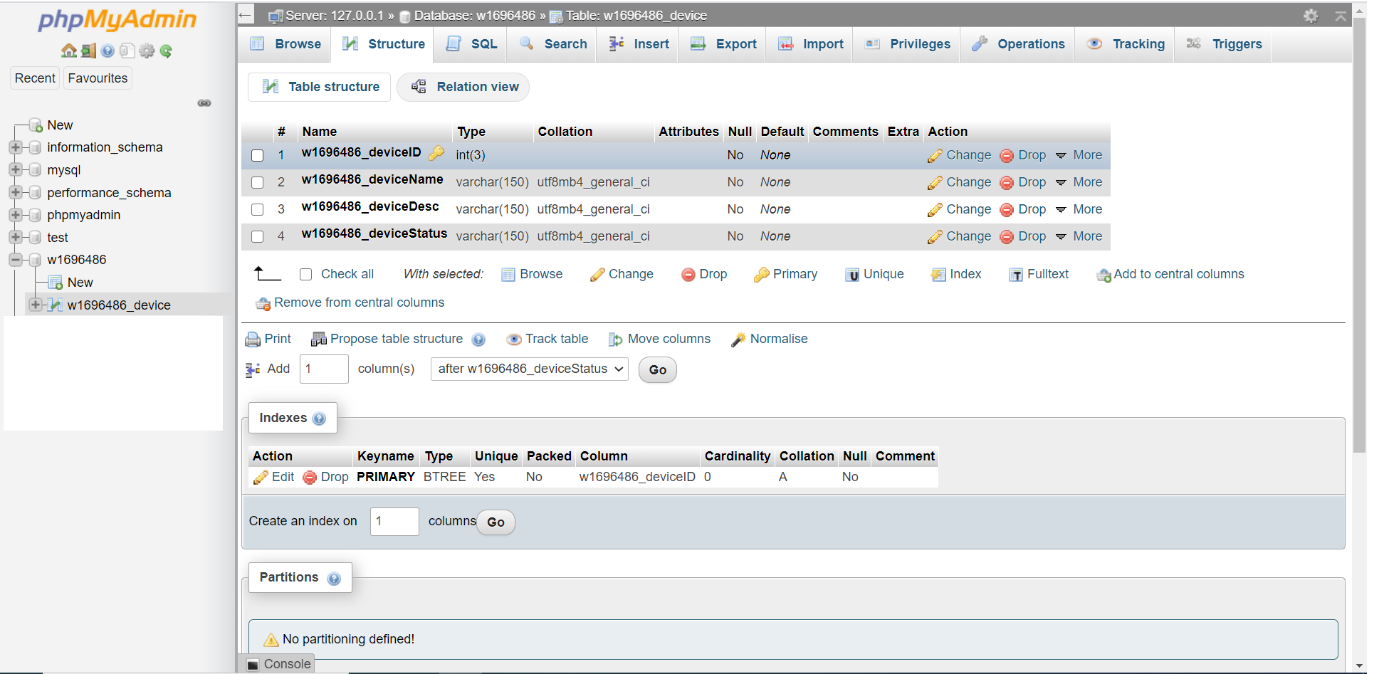
w1696486\_deviceName VARCHAR(150) NOT NULL,

w1696486\_deviceDesc VARCHAR(150) NOT NULL,

w1696486\_deviceStatus VARCHAR(150) NOT NULL,

CONSTRAINT d\_did\_PK PRIMARY KEY (w1696486\_deviceID)

);



**SQL code for the table w1696486\_Visual\_Device**

CREATE TABLE w1696486\_Visual\_Device

(

w1696486\_deviceID INT (3),

w1696486\_frameFlag BOOLEAN,

w1696486\_frameBrand VARCHAR(35),

w1696486\_frameModel VARCHAR(35),

w1696486\_lensFlag BOOLEAN,

w1696486\_lensSeriaIID VARCHAR(60),

w1696486\_lensVisionType VARCHAR(150),

w1696486\_lensTValue VARCHAR (60),

w1696486\_lensThinnessLevel VARCHAR (60),

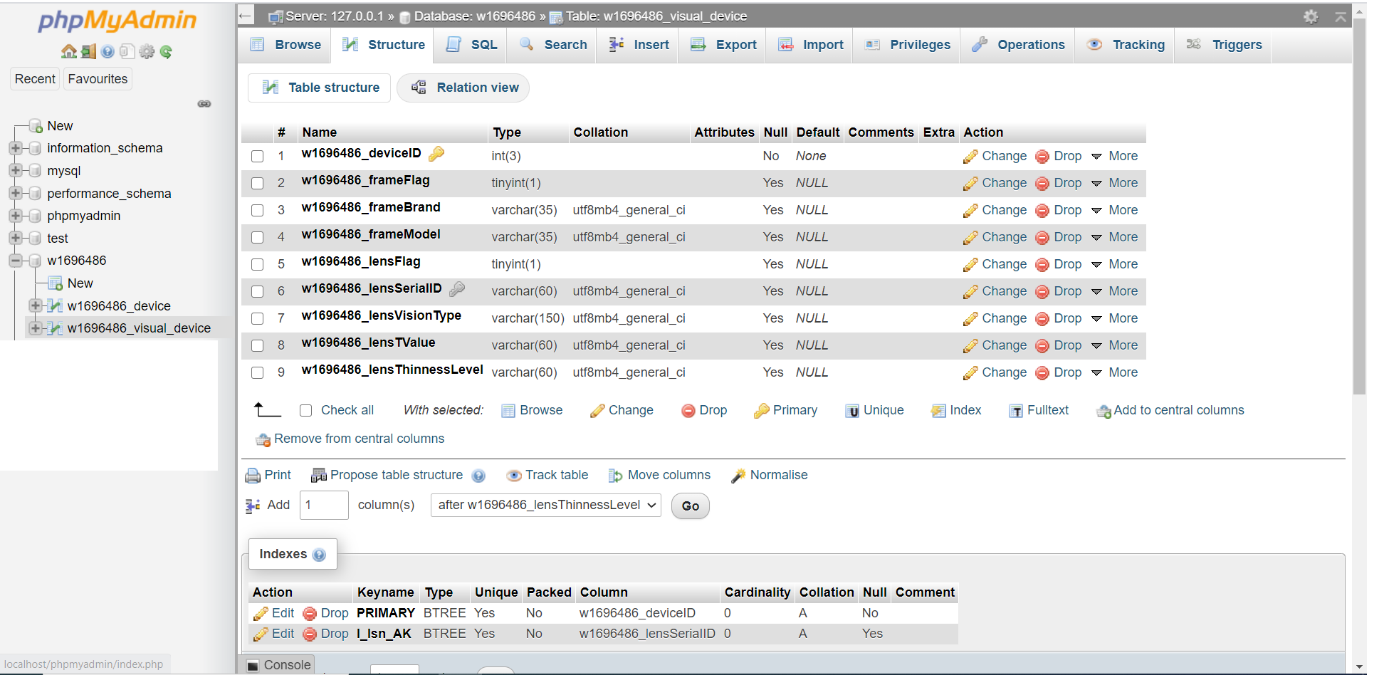
CONSTRAINT vd\_vdid\_PK PRIMARY KEY (w1696486\_deviceID),

CONSTRAINT l\_lsn\_AK UNIQUE KEY (w1696486\_lensSeriaIID),

CONSTRAINT vd\_vdid\_FK FOREIGN KEY (w1696486\_deviceID)

REFERENCES w1696486\_Device(w1696486\_deviceID)

);



**SQL code for the table w1696486\_Hearing\_Device**

CREATE TABLE w1696486\_Hearing\_Device

(

w1696486\_deviceID INT(3),

w1696486\_hdProducer VARCHAR (45),

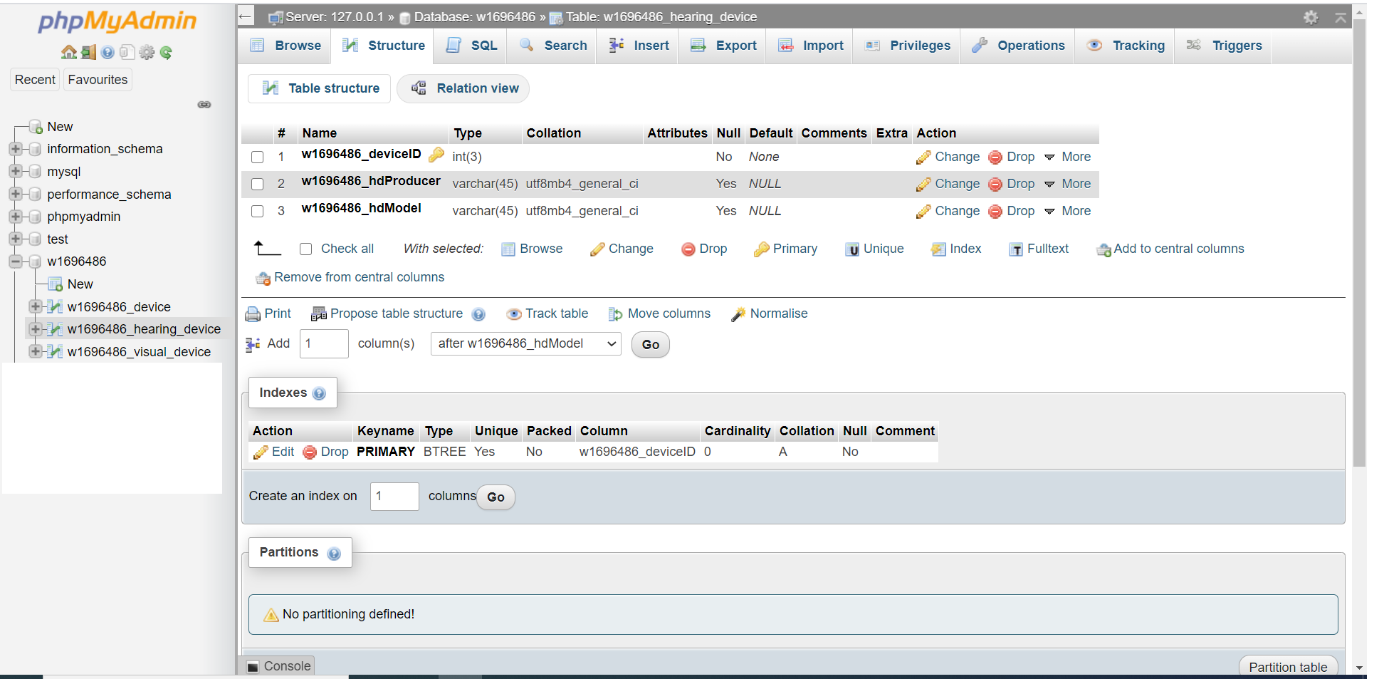
w1696486\_hdModel VARCHAR (45),

CONSTRAINT hd\_hdid\_PK PRIMARY KEY (w1696486\_deviceID),

CONSTRAINT hd\_hdid\_FK FOREIGN KEY (w1696486\_deviceID)

REFERENCES w1696486\_Device(w1696486\_deviceID)

);



**TASK - 8**

**Inserting Data Into the Tables**

**Inserting data in to the table w1696486\_Device**

INSERT INTO

w1696486\_Device

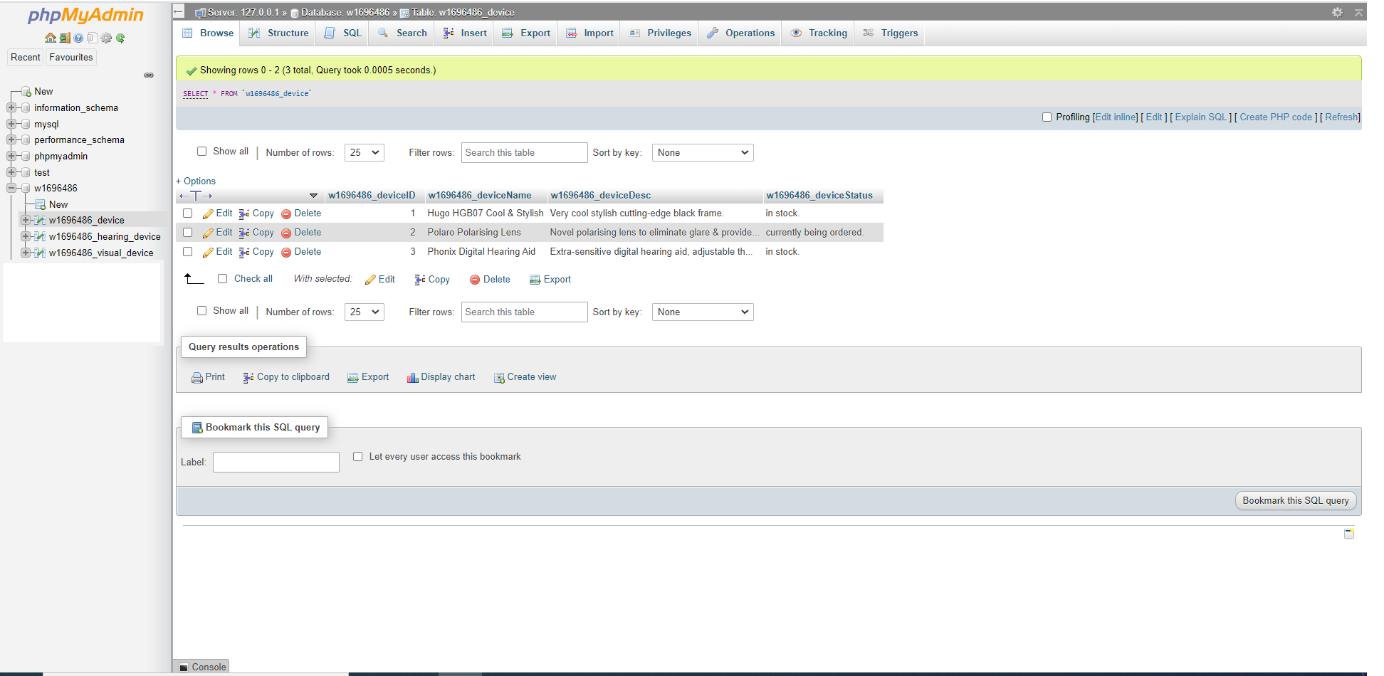
(w1696486\_deviceID, w1696486\_deviceName,w1696486\_deviceDesc, w1696486\_deviceStatus)

VALUES

(1, 'Hugo HGB07 Cool & Stylish ', 'Very cool stylish cutting-edge black frame.', 'in stock.'),

(2, 'Polaro Polarising Lens', 'Novel polarising lens to eliminate glare & provide UV protection.', 'currently being ordered.'),

(3, 'Phonix Digital Hearing Aid', 'Extra-sensitive digital hearing aid, adjustable through App on IoS.', 'in stock.');



**Inserting data in to the table w1696486\_Visual\_Device**

INSERT INTO

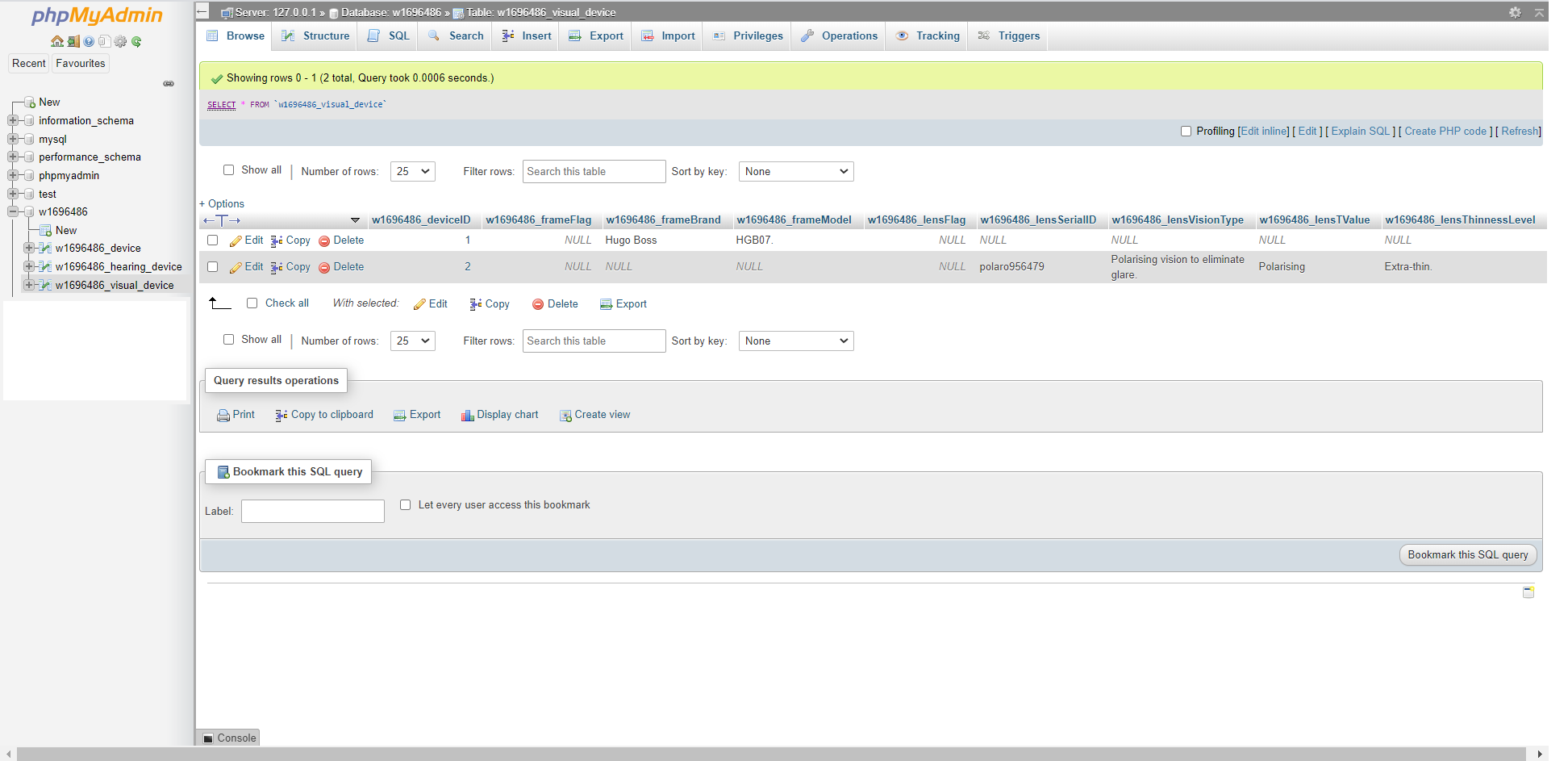
w1696486\_Visual\_Device(w1696486\_deviceID, w1696486\_frameFlag,

w1696486\_frameBrand, w1696486\_frameModel, w1696486\_lensFlag, w1696486\_lensSeriaIID,

w1696486\_lensVisionType, w1696486\_lensTValue, w1696486\_lensThinnessLevel)

VALUES(1, NULL, 'Hugo Boss', 'HGB07.', NULL, NULL, NULL, NULL, NULL),

(2, NULL, NULL, NULL, NULL, 'polaro956479', 'Polarising vision to eliminate glare.', 'Polarising', 'Extra-thin.');



**Inserting data in to the table w1696486\_Hearing\_Device**

INSERT INTO

w1696486\_Hearing\_Device(w1696486\_deviceId, w1696486\_hdProducer, w1696486\_hdModel)

VALUES (3, 'Phonix.', 'phonix007.');

