|  |
| --- |
| ITDA310 Project |
| Advanced Database Systems: Final Deliverable |
| W7MSRF657 |

|  |
| --- |
| Francois@bookwise.co.za  Francois Etienne Cloete |

Contents

[Deliverable 1 2](#_Toc42507613)

[ER Diagram 2](#_Toc42507614)

[Data Dictionary 3](#_Toc42507615)

[Deliverable 2: Logical Data Model 6](#_Toc42507616)

[Derived Relations (Connolly and Begg, 2014) 6](#_Toc42507617)

[Normalization 7](#_Toc42507618)

[Unnormalized form 7](#_Toc42507619)

[first normal form (removing repeating groups) (Connolly and Begg, 2014) 8](#_Toc42507620)

[Second Normal Form (Removing partial dependencies) (Connolly and Begg, 2014) 9](#_Toc42507621)

[Third Normal Form (Removing Transitive Dependencies) (Connolly and Begg, 2014) 11](#_Toc42507622)

[Checking integrity Constraints 13](#_Toc42507623)

[Required Data (Connolly and Begg, 2014) 13](#_Toc42507624)

[Domains 15](#_Toc42507625)

[Multiplicity 15](#_Toc42507626)

[Entity integrity 15](#_Toc42507627)

[Referential Integrity 15](#_Toc42507628)

[General Cases (relating to all tables unless otherwise stated in the global data model): 15](#_Toc42507629)

[Global Data Model 16](#_Toc42507630)

[Global Entity Relationship Model 20](#_Toc42507631)

[Deliverable 3: Physical Database Design 21](#_Toc42507632)

[Base Relations Design and General Integrity Constraints 21](#_Toc42507633)

[Pseudo Data 27](#_Toc42507634)

[User Views 30](#_Toc42507635)

[Security Mechanisms 31](#_Toc42507636)

[Deliverable 4: Advanced Query Processing 38](#_Toc42507637)

[Cursor 38](#_Toc42507638)

[Stored Procedure 39](#_Toc42507639)

[Function 40](#_Toc42507640)

[Triggers 41](#_Toc42507641)

[Deliverable 5: Database Interface Program 45](#_Toc42507642)

[Interface 45](#_Toc42507643)

[References 47](#_Toc42507644)

# Deliverable 1

## ER Diagram

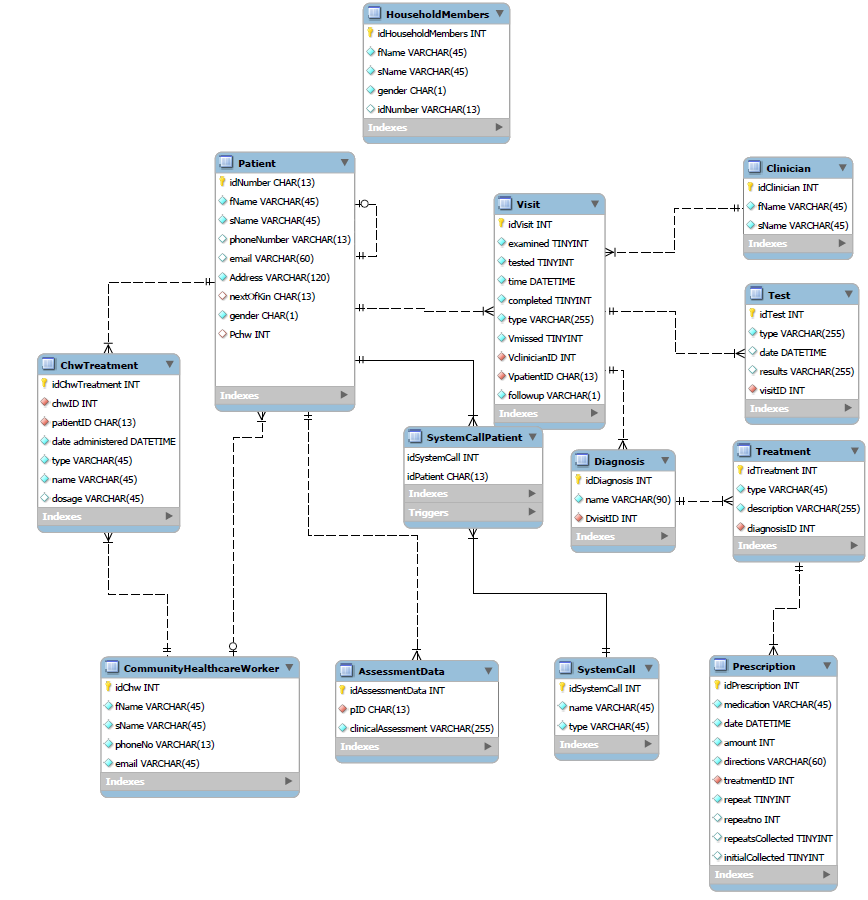


Figure : Conceptual Entity Relationship Diagram

## Data Dictionary

For a higher resolution version of the data dictionary, please open ITDA310 DD3

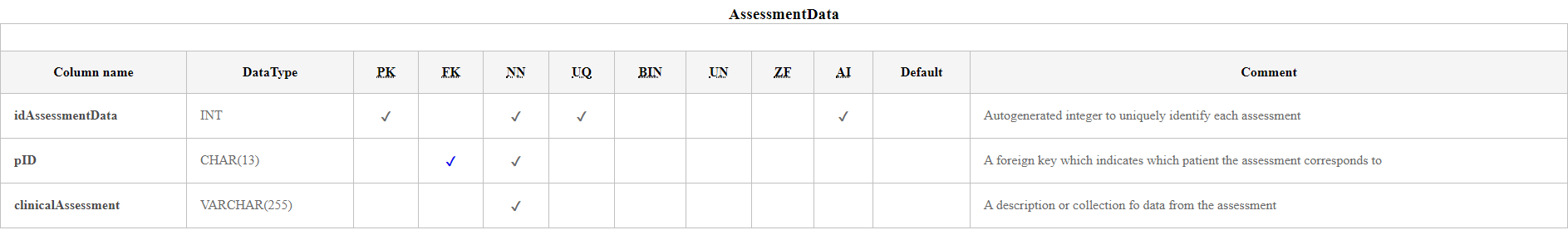


Figure : AssessmentData Entity (Castro, 2018)

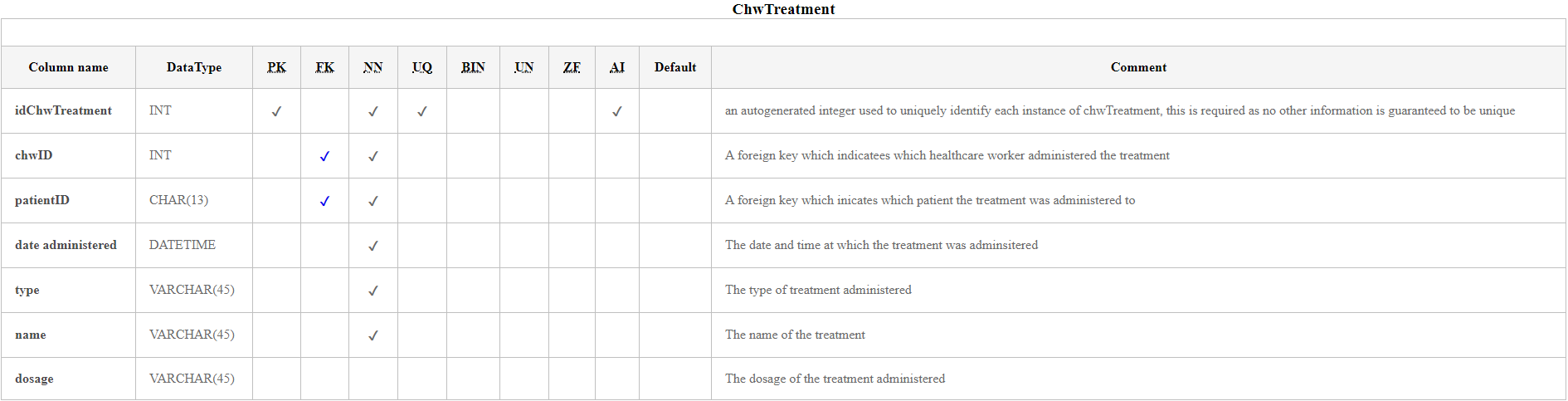


Figure : ChwTreatment Entity (Castro, 2018)

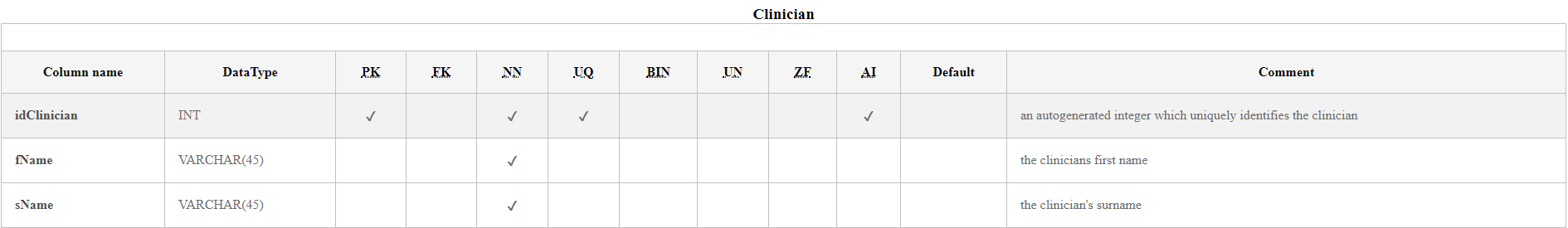


Figure : Clinician Entity (Castro, 2018)

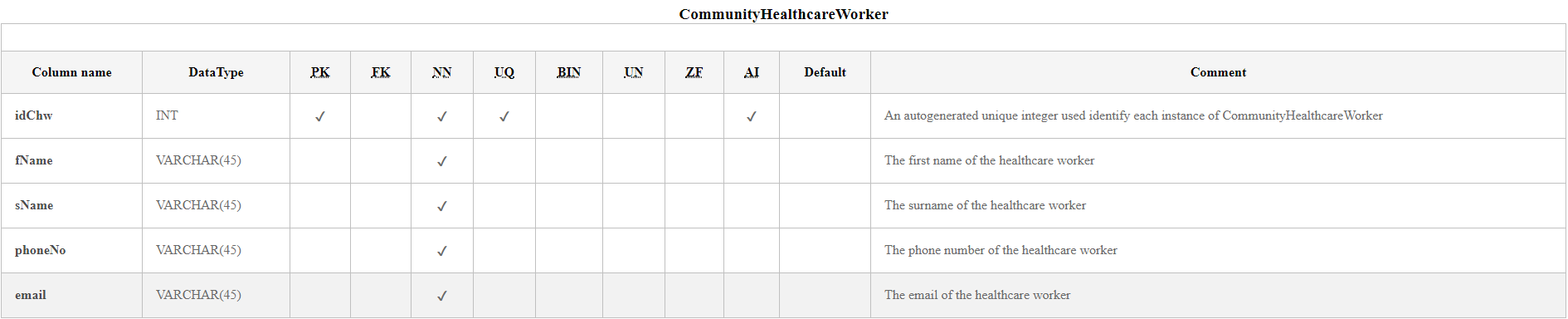


Figure : Community Healthcare Worker Entity (Castro, 2018)

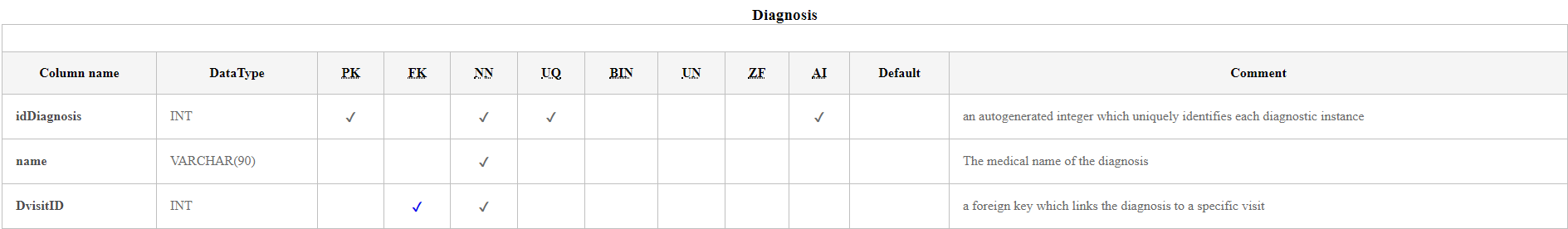


Figure : Diagnosis Entity (Castro, 2018)

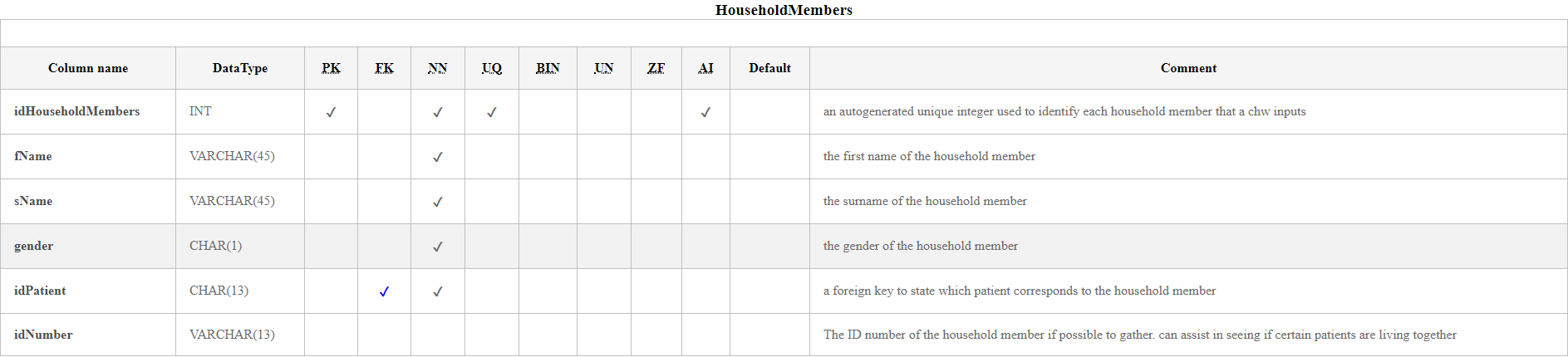


Figure : HouseholdMembers Entity (Castro, 2018)

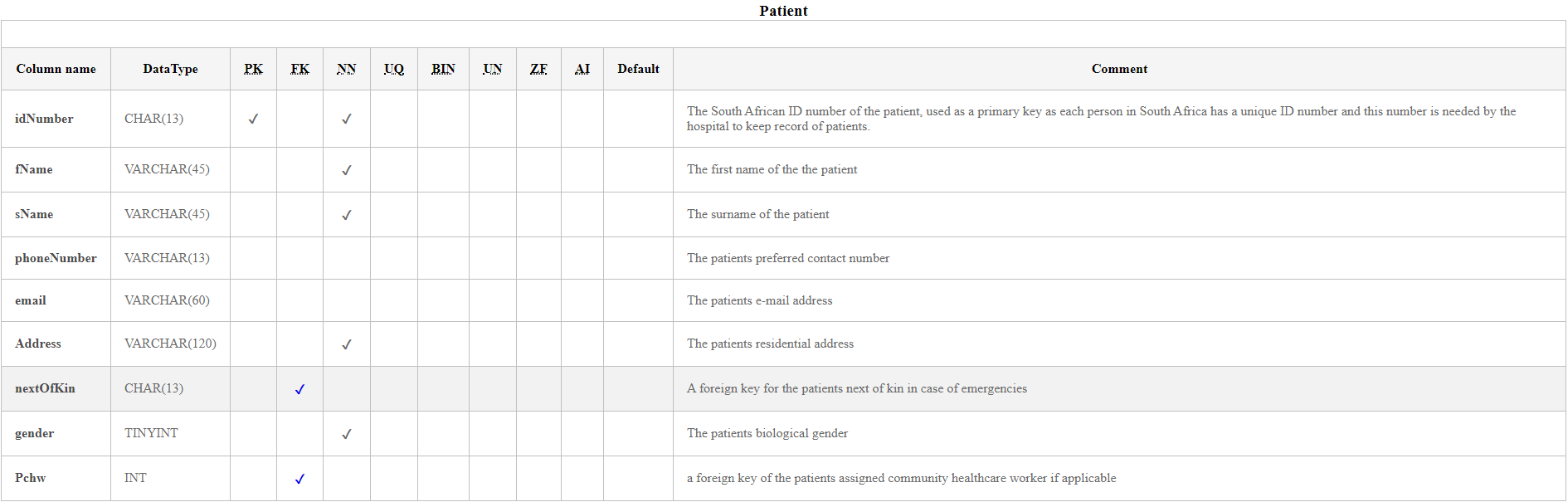


Figure : Patient Entity (Castro, 2018)

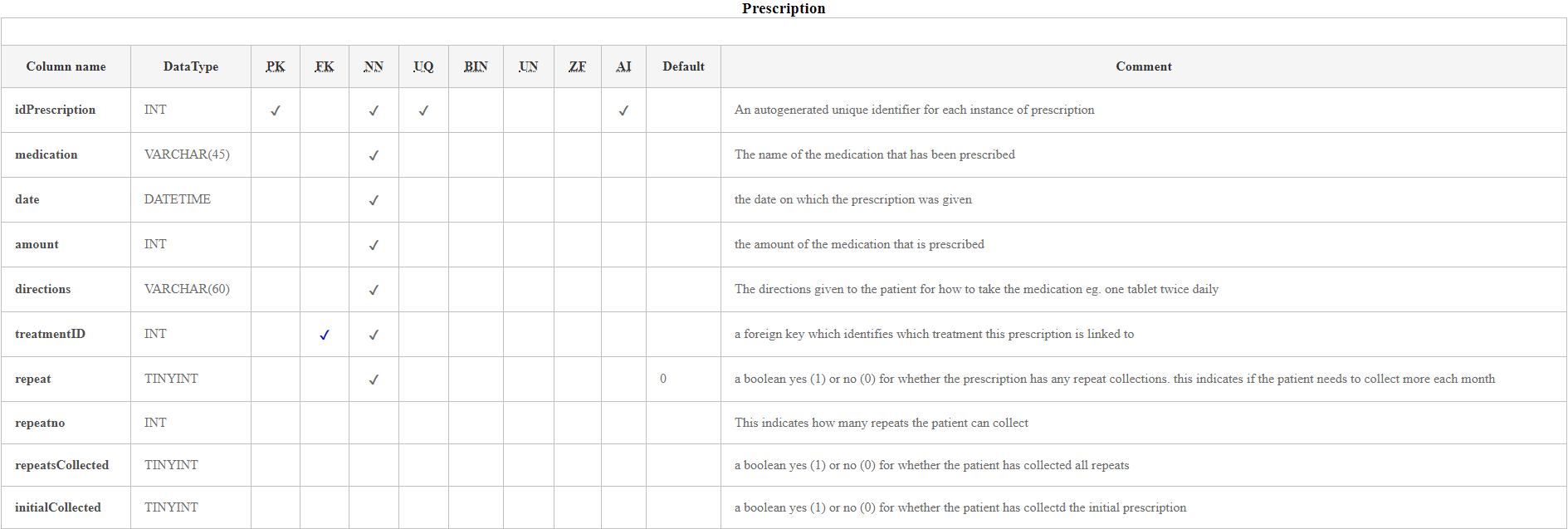


Figure : Prescription Entity (Castro, 2018)

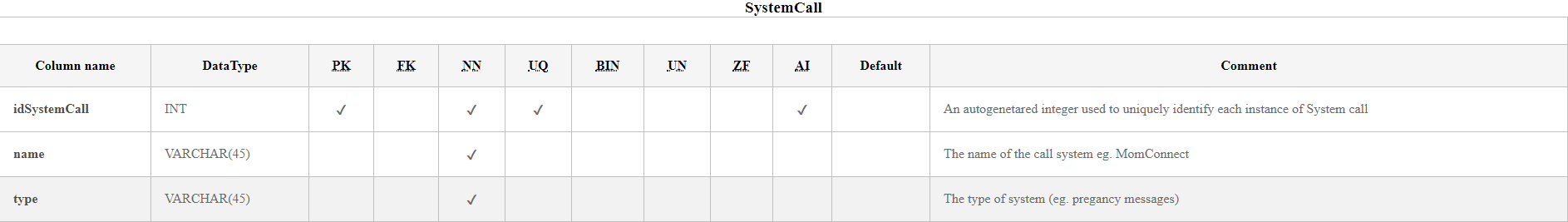


Figure : SystemCall Entity (Castro, 2018)

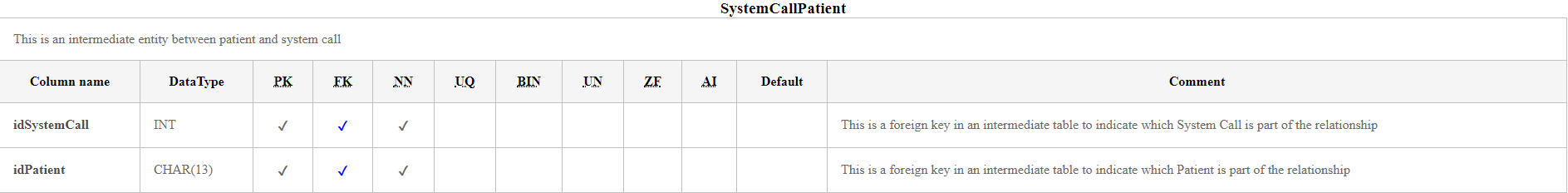


Figure : SystemCallPatient Entity (Castro, 2018)

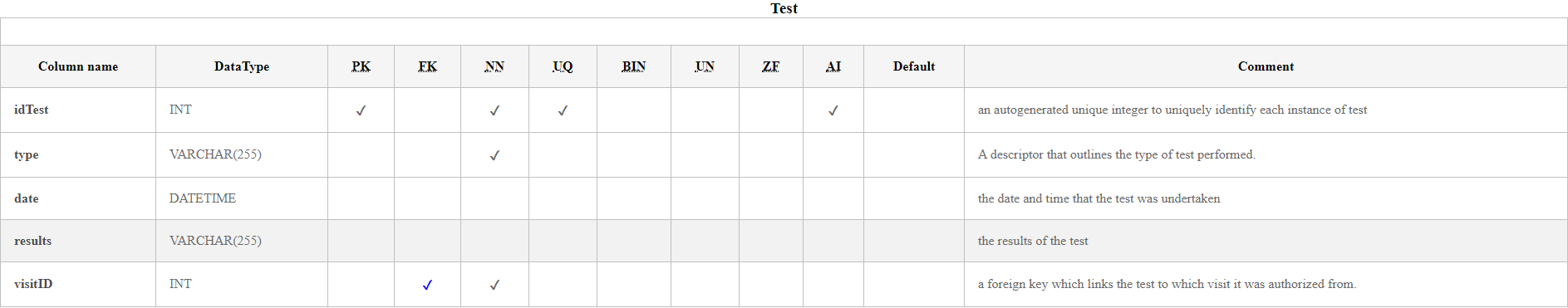


Figure : Test Entity (Castro, 2018)

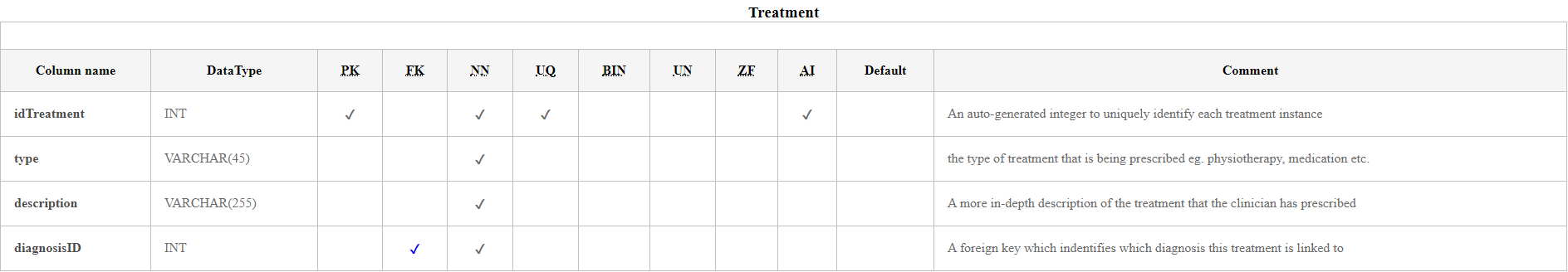


Figure : Treatment Entity (Castro, 2018)

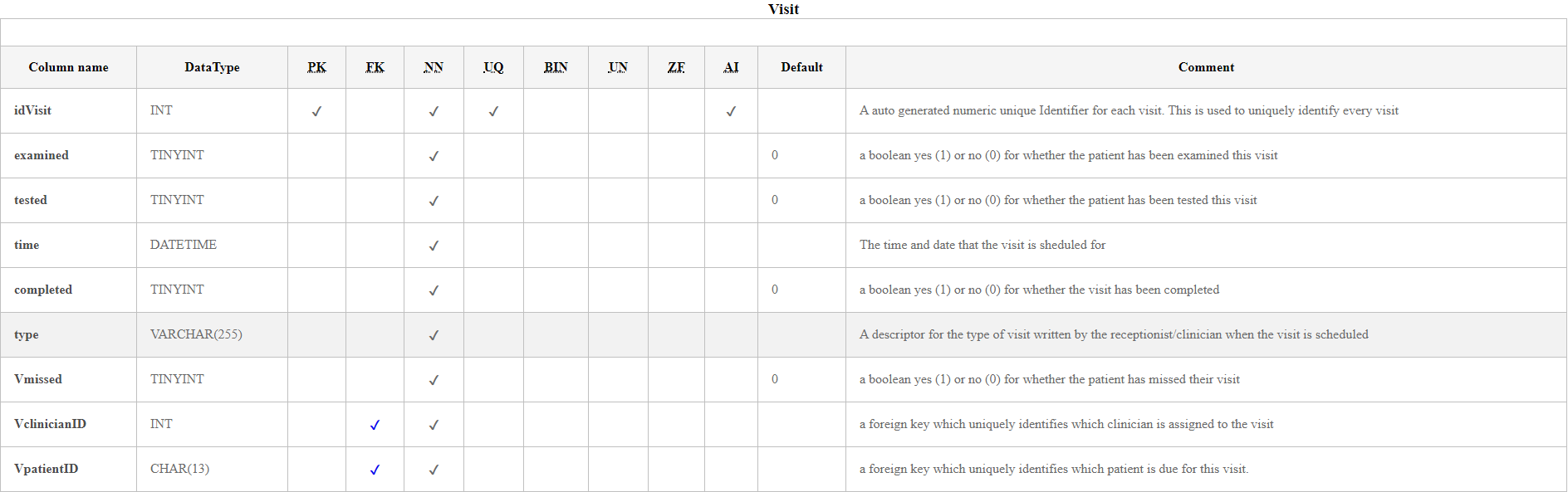


Figure : Treatment Entity (Castro, 2018)

# Deliverable 2: Logical Data Model

## Derived Relations (Connolly and Begg, 2014)

* Each Patient has one or more visits – each visit has one and only one patient (1:\*)
* Each visit has one and only one clinician – each clinician has zero or more visits (1:\*)
* Each visit results in zero or more lab tests – Each lab test is allocated to one and only one visit (1:\*)
* Each visit results in zero or more diagnoses – Each instance of Diagnosis is assigned to one and only one visit (1:\*)
* Each diagnosis has one or more treatments – each treatment is assigned to only one diagnosis (1:\*)
* Each treatment has zero or more prescriptions – each prescription is allocated to only one treatment. (1:\*)
* Each patient can have 0 or more system calls – each system call can have 0 or more patients (\*:\*)
* Each community healthcare worker looks after 1 or more patients - each patient is assigned one and only one community healthcare worker (1:\*)
* Each patient has zero or more household members – each instance of household member lives with one or more patients (\*:\*)
* Each Patient has one and only one next of kin (registered as another patient) (1:1)
* Each Patient has 0 or more chwTreatments – each community healthcare worker administers zero or more treatments (\*:\*)
* Each community Healthcare Worker Capture one or more sets of Assessment Data for Each Patient – Each set of assessment data is captured by one and only one community healthcare worker (1:\*)

## Normalization

### Unnormalized form

* (PK)PidNumber
* Pfname
* Psname
* Pphonenumber
* Pemail
* Paddress
* Pgender
* visitID
* Vexamined
* Vtested
* Vtime
* Vcompleted
* Vmissed
* Vtype
* ClinicianID
* Cfname
* Csname
* Testid
* Ttype
* Tdate
* Tresults
* DiagnosisID
* Dname
* TreatmentID
* Ttype
* Tdesc
* Medication
* Mdate
* Mamount
* Mdirections
* Mrepeat
* MrepeatNo
* Mrepeatscollected
* Minitialcollected
* SystemID
* SysName
* SysType
* idChw
* fname
* sname
* phoneNo
* email
* treatmentType
* treatmentDate
* treatmentName
* dosage
* hfname
* hsname
* hgender
* hidno
* idAssessmentData
* pID
* clinicalAssessment

### first normal form (removing repeating groups) (Connolly and Begg, 2014)

* (PK)PidNumber
* Pfname
* Psname
* Pphonenumber
* Pemail
* Paddress
* Pgender
* (FK)PidNumber
* (FK)systemID
* (PK)visitID
* Vexamined
* Vtested
* Vtime
* Vcompleted
* Vmissed
* Vtype
* (FK)PidNumber
* (FK)ClinicianID
* (PK)ClinicianID
* Cfname
* Csname
* (PK)Testid
* Ttype
* Tdate
* Tresults
* (FK)VisitID
* DiagnosisID
* Dname
* (FK)visitID
* TreatmentID
* Ttype
* Tdesc
* (FK)diagnosisID
* (PK)Medicationid
* Mdate
* Mamount
* Mdirections
* Mrepeat
* MrepeatNo
* Mrepeatscollected
* Minitialcollected
* (FK)treatmentID
* (PK)SystemID
* SysName
* SysType
* (PK)idChw
* fname
* sname
* phoneNo
* email
* treatmentType
* treatmentDate
* treatmentName
* dosage
* (PK)householdMemberID
* hfname
* hsname
* hgender
* hidno
* (FK)PidNumber
* (PK)idAssessmentData
* (FK)pID
* clinicalAssessment

### Second Normal Form (Removing partial dependencies) (Connolly and Begg, 2014)

* (PK)PidNumber
* Pfname
* Psname
* Pphonenumber
* Pemail
* Paddress
* Pgender
* (FK)nextOfKin
* (PK)visitID
* Vexamined
* Vtested
* Vtime
* Vcompleted
* Vmissed
* Vtype
* (FK)PidNumber
* (FK)ClinicianID
* (PK)ClinicianID
* Cfname
* Csname
* (PK)Testid
* Ttype
* Tdate
* Tresults
* (FK)VisitID
* DiagnosisID
* Dname
* (FK)visitID
* TreatmentID
* Ttype
* Tdesc
* (FK)diagnosisID
* (PK)Medicationid
* Mdate
* Mamount
* Mdirections
* Mrepeat
* MrepeatNo
* Mrepeatscollected
* Minitialcollected
* (FK)treatmentID
* (PK)SystemID
* SysName
* SysType
* (FK)SystemID
* (FK)patientID
* (PK)idChw
* fname
* sname
* phoneNo
* email
* chwTreatmentID
* treatmentType
* treatmentDate
* treatmentName
* dosage
* (FK)idChw
* (FK)PidNumber
* (PK)householdMemberID
* hfname
* hsname
* hgender
* hidno
* (FK)householdMemberID
* (FK)PidNumber
* (PK)idAssessmentData
* (FK)pID
* clinicalAssessment

### Third Normal Form (Removing Transitive Dependencies) (Connolly and Begg, 2014)

* (PK)PidNumber
* Pfname
* Psname
* Pphonenumber
* Pemail
* Paddress
* Pgender
* (FK)nextOfKin
* (PK)visitID
* Vexamined
* Vtested
* Vtime
* Vcompleted
* Vmissed
* Vtype
* (FK)PidNumber
* (FK)ClinicianID
* (PK)ClinicianID
* Cfname
* Csname
* (PK)Testid
* Ttype
* Tdate
* Tresults
* (FK)VisitID
* DiagnosisID
* Dname
* (FK)visitID
* TreatmentID
* Ttype
* Tdesc
* (FK)diagnosisID
* (PK)Medicationid
* Mdate
* Mamount
* Mdirections
* Mrepeat
* MrepeatNo
* Mrepeatscollected
* Minitialcollected
* (FK)treatmentID
* (PK)SystemID
* SysName
* SysType
* (FK)SystemID
* (FK)patientID
* (PK)idChw
* fname
* sname
* phoneNo
* email
* chwTreatmentID
* treatmentType
* treatmentDate
* treatmentName
* dosage
* (FK)idChw
* (FK)PidNumber
* (PK)householdMemberID
* hfname
* hsname
* hgender
* hidno
* (FK)householdMemberID
* (FK)PidNumber
* (PK)idAssessmentData
* (FK)pID
* clinicalAssessment

## Checking integrity Constraints

### Required Data (Connolly and Begg, 2014)

Fields that are required (separated by table):

* Assessment Data
  + idAssessmentData
  + pID
  + clinicalAssessment
* ChwTreatment
  + idChwTreatment
  + chwID
  + patientID
  + date administered
  + type
  + name
* Clinician
  + idClinician
  + fName
  + sName
* CommunityHealthcareWorker
  + idChw
  + fName
  + sName
  + phoneNo
  + email
* Diagnosis
  + idDiagnosis
  + name
  + DvisitID
* Household
  + idPatient
  + idMember
* HouseholdMembers
  + idHouseholdMembers
  + fName
  + sName
  + gender
* Patient
  + idNumber
  + fName
  + sName
  + address
  + gender
* Prescription
  + idPrescription
  + medication
  + date
  + amount
  + directions
  + treatmentID
  + repeat
* SystemCall
  + idSystemCall
  + name
  + type
* SystemCallPatient
  + idSystemCall
  + idPatient
* Test
  + idTest
  + type
  + visitID
* Treatment
  + idTreatment
  + type
  + description
  + diagnosisID
* Visit
  + idVisit
  + examined
  + tested
  + time
  + completed
  + type
  + vMissed
  + vClinicianID
  + vPatientID

### Domains

See Global Data Model

### Multiplicity

See the relationships outlined in Derived relationships above. Multiplicity is outlined at the end of each relationship. There is no cap on one to many relationships to help assure Scalability. (Connolly and Begg, 2014)

### Entity integrity

Every entity has its Primary key set to being not null to ensure entity integrity (Connolly and Begg, 2014)

### Referential Integrity

### General Cases (relating to all tables unless otherwise stated in the global data model):

(Connolly and Begg, 2014)

* Case 1: insert tuple into child relation:
  + To ensure referential integrity, check that the foreign key attributes of the new tuple are set to null if possible or to the value of an existing tuple in the parent entity
* Case 2: delete tuple from the child relation
  + If a tuple in the child relation is deleted, referential integrity is unaffected
* Case 3: update the foreign key of the child tuple
  + This is similar to case 1. To ensure referential integrity, check whether the updated foreign key is set to null or to the value of an existing staff tuple
* Case 4: insert tuple into parent relation:
  + Inserting a tuple into the parent relation does not affect referential integrity, it simply creates a parent instance without any children. Eg. A newly hired clinician who has not seen any patient yet.
* Case 5: Delete tuple from parent relation:
  + a tuple of a parent relation is deleted, referential integrity is lost if there exists a child tuple referencing the deleted parent tuple. In this case there are several strategies to consider:
    - NO ACTION – this prevents the deletion of a parent tuple if it has child tuples
    - CASCADE – When the parent tuple is deleted, all referring child tuples will also be deleted
    - SET NULL – When a parent tuple is deleted, the foreign key values in child tuples are set to null
    - SET DEFAULT - When a parent tuple is deleted, the foreign key value in any child tuples will be set to a default value
    - NO CHECK – when a parent tuple is deleted, nothing is done to child tuples to ensure referential integrity.
  + For the use case in this database, a combination of NO ACTION and CASCADE will be use depending on the relationships in question. NO ACTION will ensure that child records need to be manually deleted if seen as irrelevant or will need to be manually updated to a new parent relation. Cascading deletes will be used sparingly and only in situations where valuable information cannot be lost through a cascade (for example, with patients and household members linking to them).
* Case 6: Updating the primary key of the parent tuple
  + If the primary key value is updated in a parent tuple, referential integrity would be lost if it has any child relations referencing the old primary key value. In this case Cascade will be used as stated above to ensure that child tuples are automatically updated to reflect the new primary key of the parent tuple

## Global Data Model

For a higher Resolution version see ITDA310 DDL.html

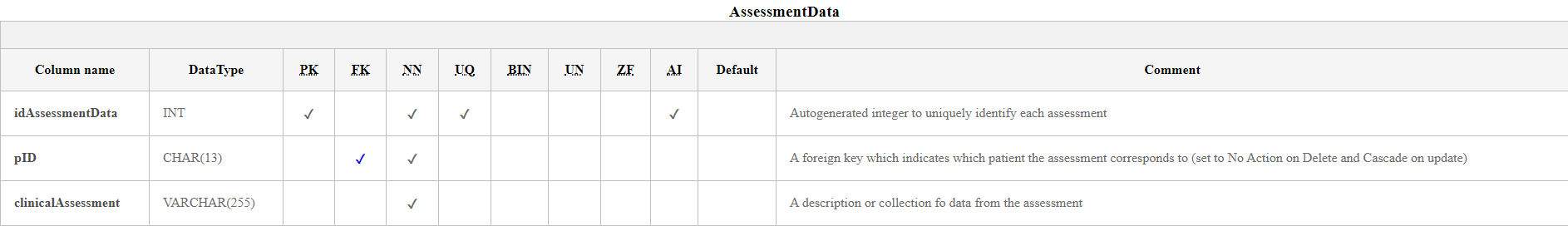


Figure : AssessmentData Entity (Castro, 2018)

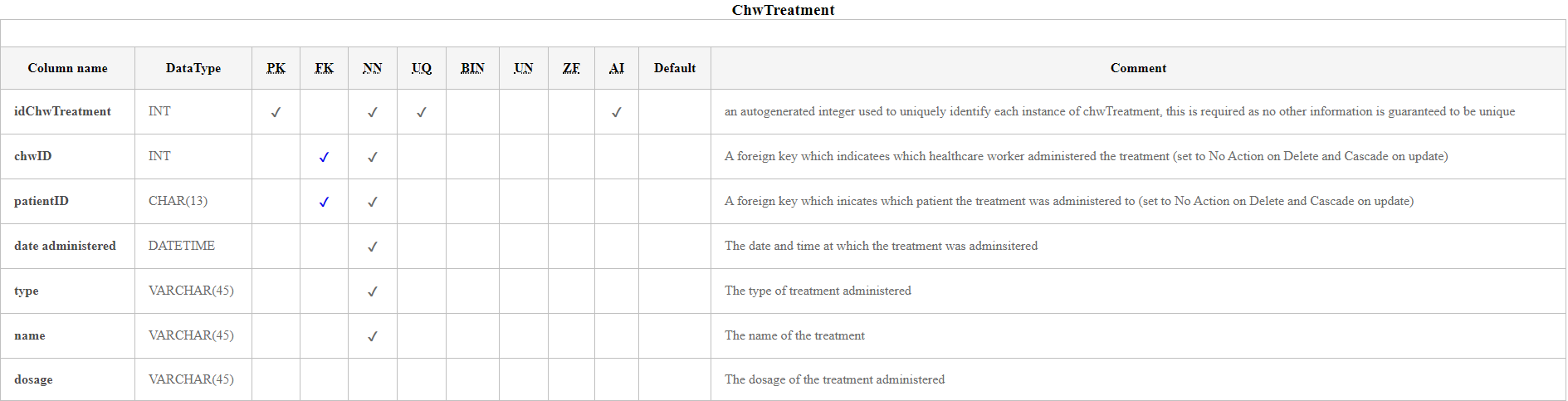


Figure :ChwTreatment Entity (Castro, 2018)

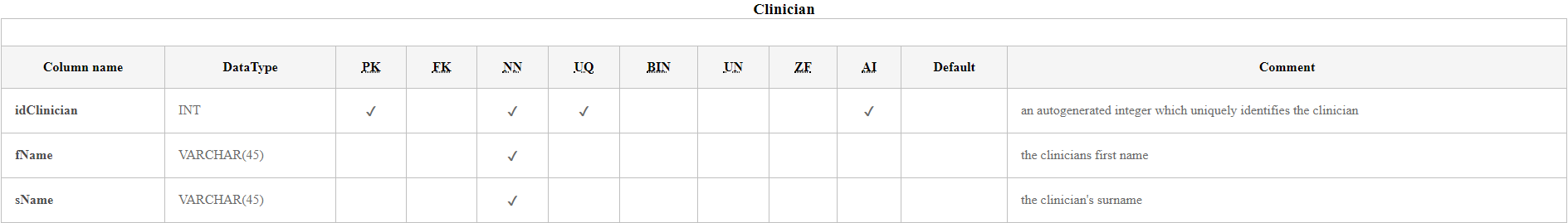


Figure : Clinician Entity (Castro, 2018)

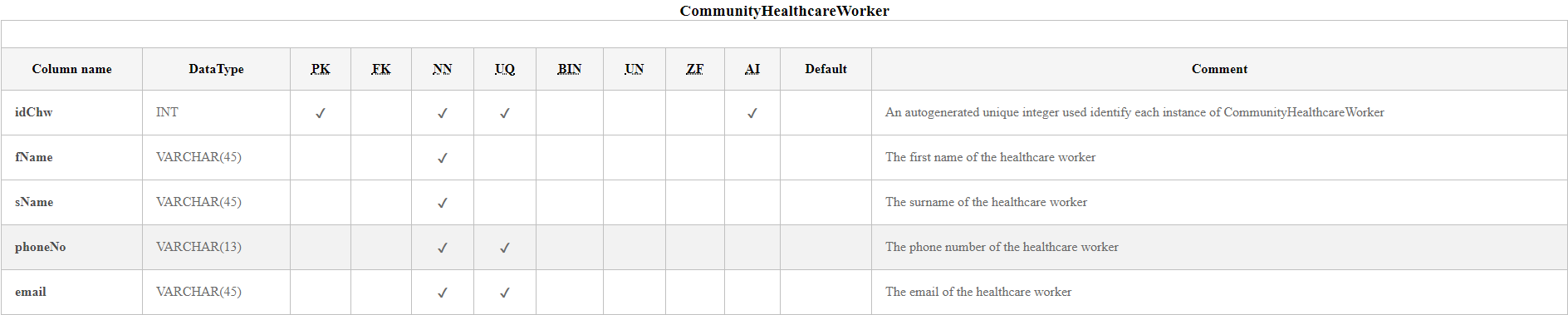


Figure : CommunityHealthcareWorker Entity (Castro, 2018)

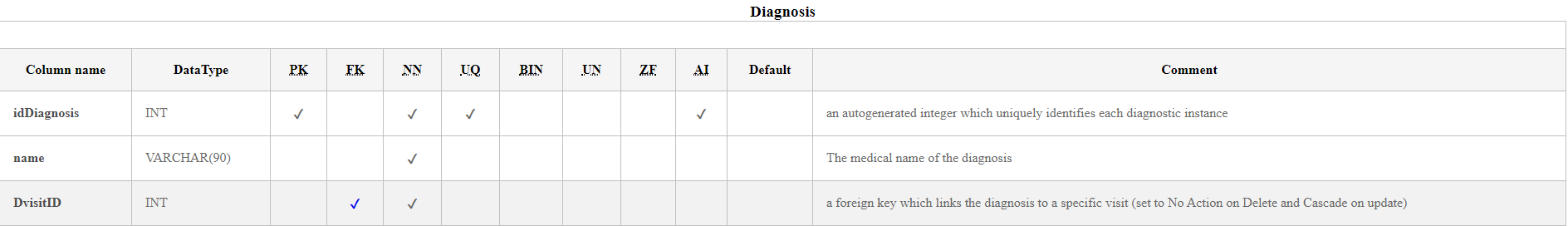


Figure : Diagnosis Entity (Castro, 2018)

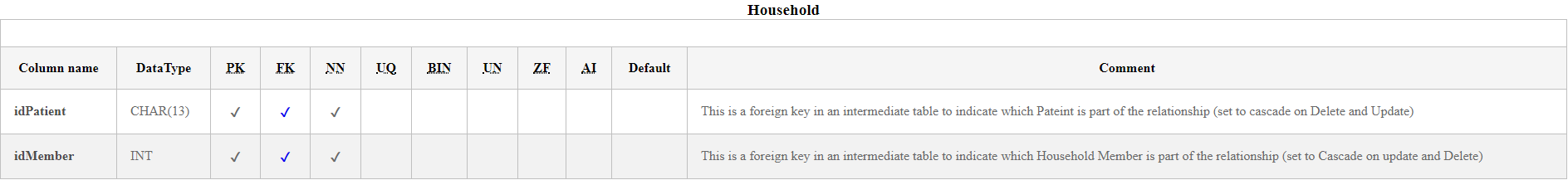


Figure : Household Entity (Castro, 2018)

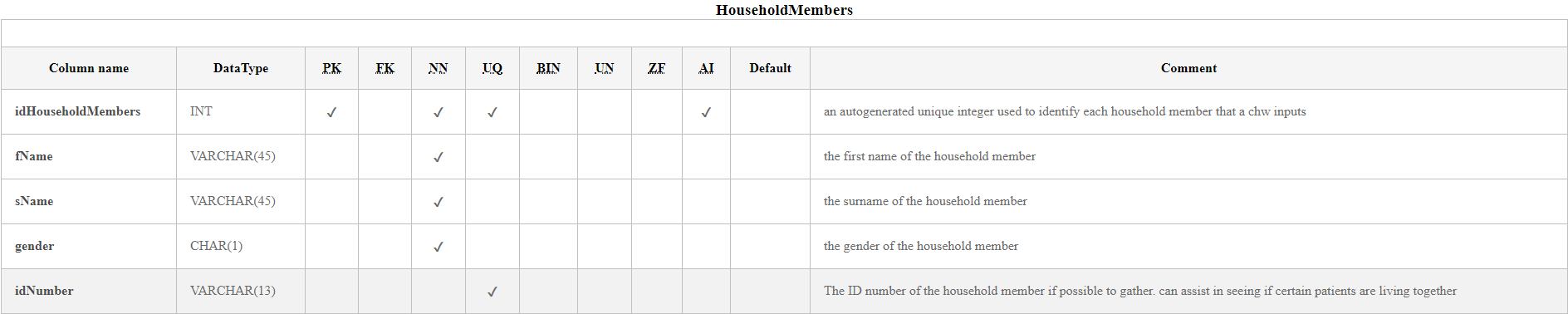


Figure : HouseholdMembers Entity (Castro, 2018)

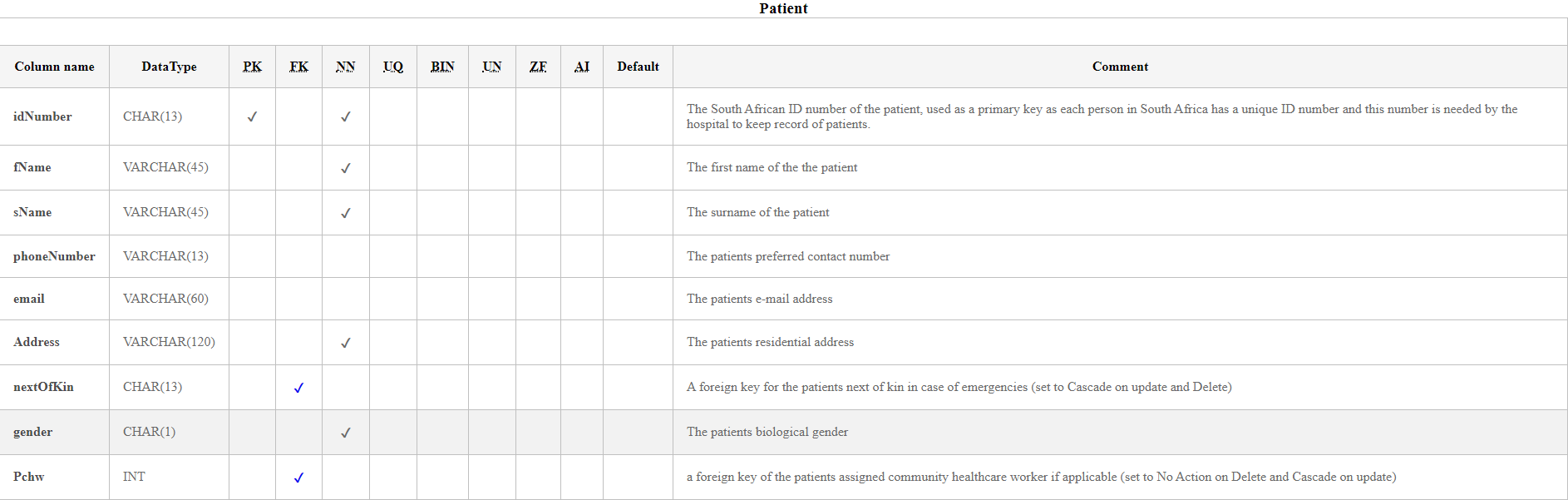


Figure : Patient Entity (Castro, 2018)

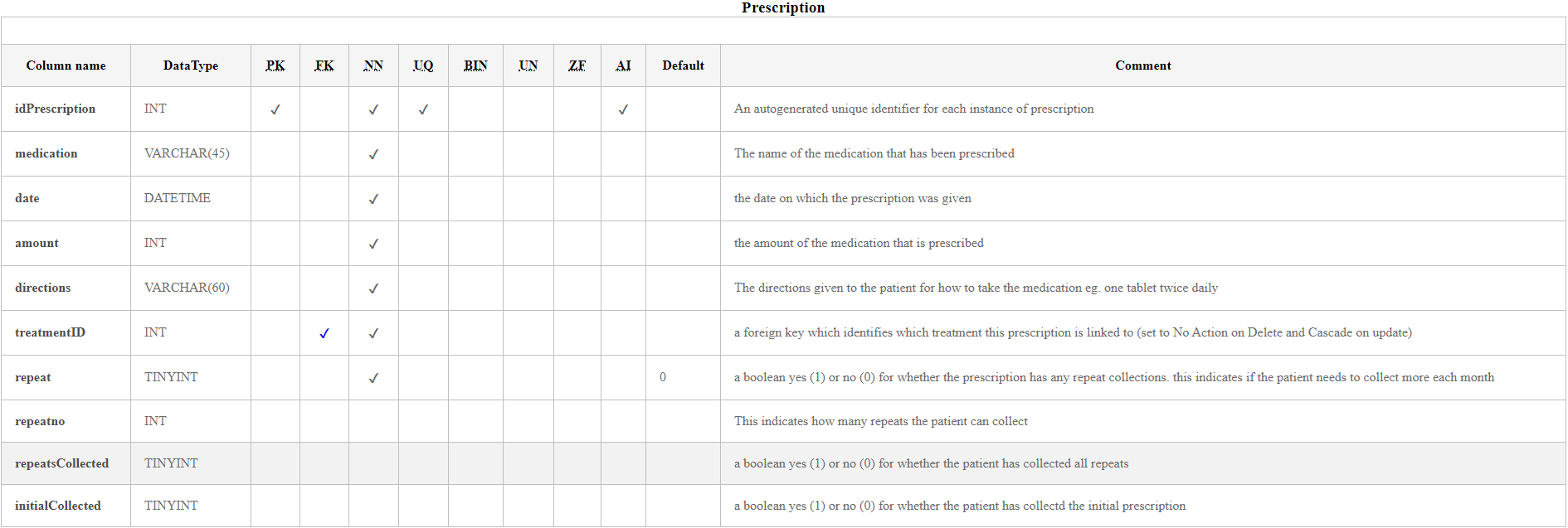


Figure : Prescription Entity (Castro, 2018)

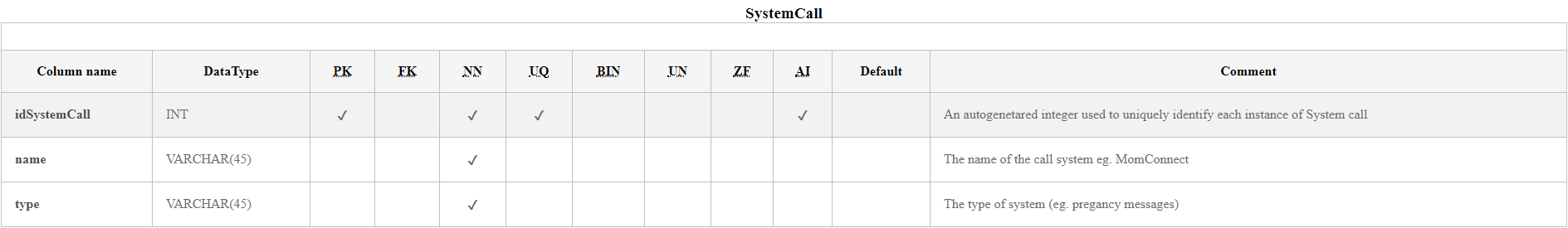


Figure : SystemCall Entity (Castro, 2018)

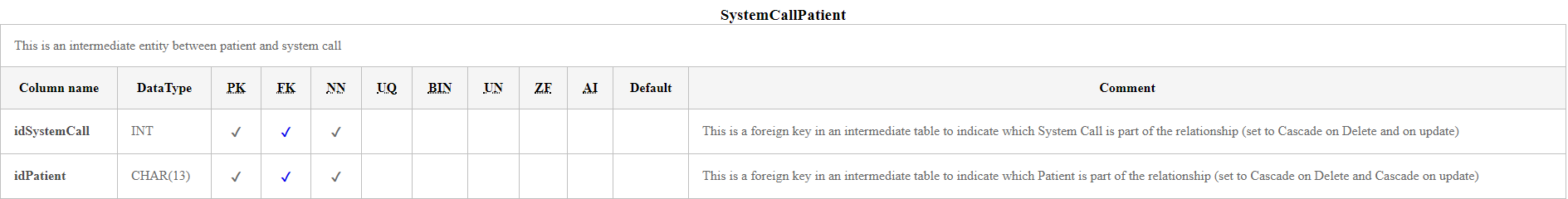


Figure :SystemCallPatient Entity (Castro, 2018)

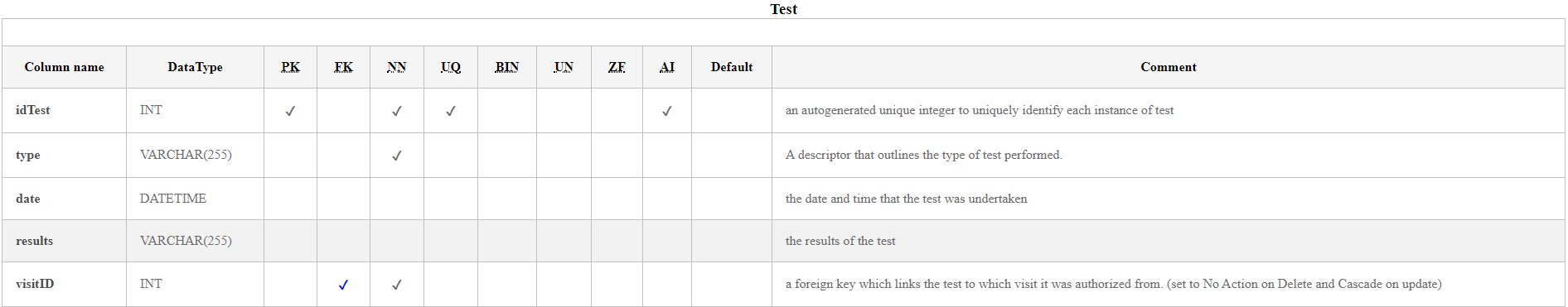


Figure : Test Entity (Castro, 2018)

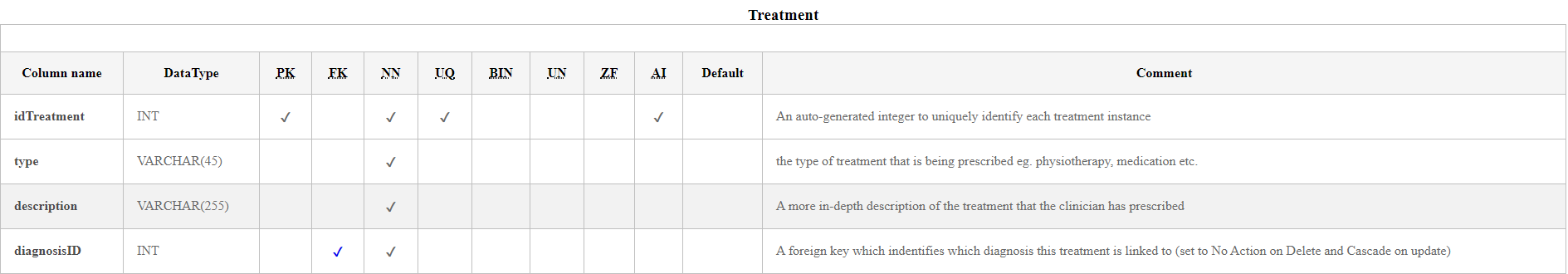


Figure : Treatment Entity (Castro, 2018)

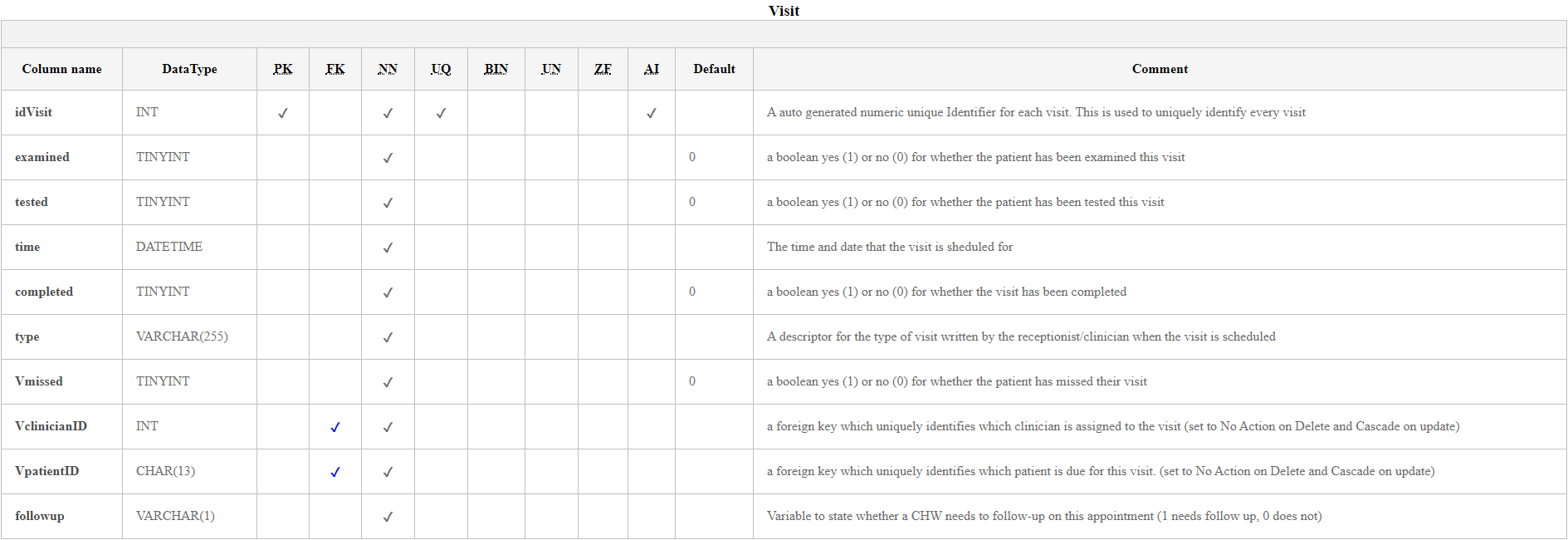


Figure : Visit Entity (Castro, 2018)

## Global Entity Relationship Model

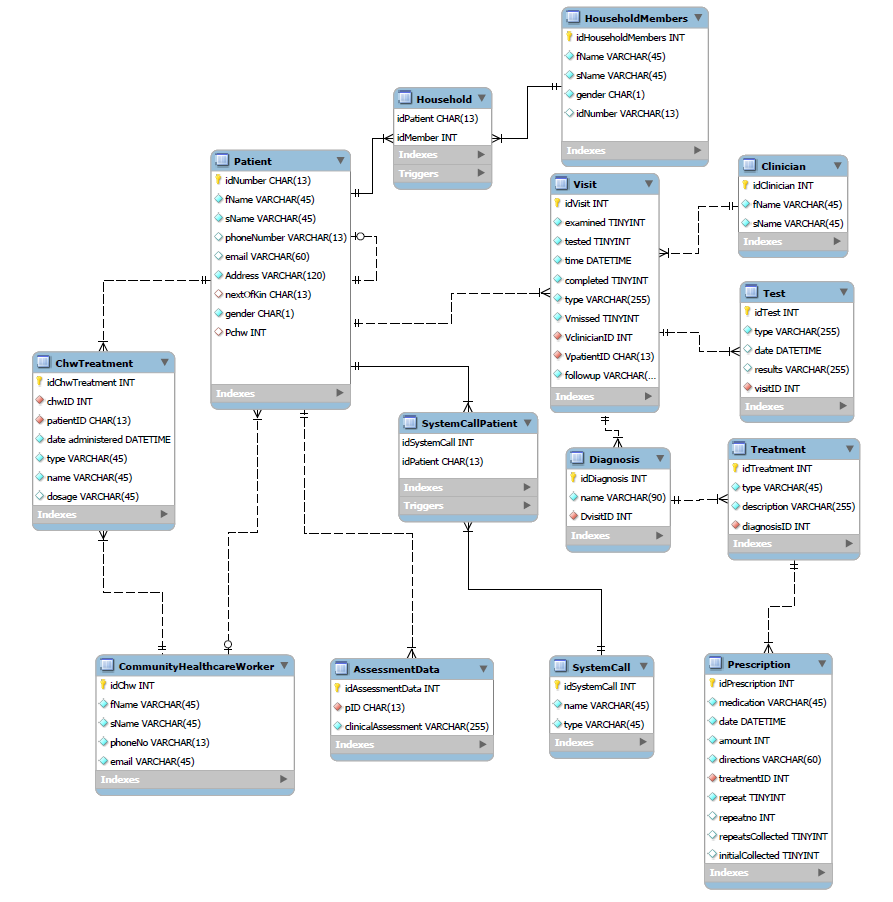


Figure : Global Entity Relationship Diagram

# Deliverable 3: Physical Database Design

## Base Relations Design and General Integrity Constraints

A screenshot of a social media post

Description automatically generated

Figure :Schema Settings for Database

A screenshot of a social media post

Description automatically generated

Figure : base relations and integrity constraints for AssessmentData

A screenshot of a social media post

Description automatically generated

Figure :Base Relations and Integrity Constraints for CommunityHealthcareWorker

A screenshot of a cell phone

Description automatically generated

Figure : Alteration to Add Missing Comment for Phone Number in Community Healthcare Worker

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for ChwTreatment

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for Clinician

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for Diagnosis

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for Household

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for HouseholdMembers

A screenshot of a social media post

Description automatically generated

Figure :Base Relations and Integrity Constraints for Patient

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for Prescription

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for System Call

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for SystemCallPatient

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for Test

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for Treatment

A screenshot of a social media post

Description automatically generated

Figure : Base Relations and Integrity Constraints for Visit

## Pseudo Data

A screenshot of a cell phone

Description automatically generated

Figure :Sample Data for AssessmentData

A screenshot of a social media post

Description automatically generated

Figure :Sample Data for CommunityHealthcareWorker

A screenshot of a cell phone

Description automatically generated

Figure :Sample Data for ChwTreatment

A screenshot of a social media post

Description automatically generated

Figure :Sample Data for Clinician

A screenshot of a cell phone

Description automatically generated

Figure ::Sample Data for Diagnosis

A screenshot of a cell phone

Description automatically generated

Figure : :Sample Data for HouseholdMembers

A screenshot of a cell phone

Description automatically generated

Figure : :Sample Data for Household

A screenshot of a cell phone

Description automatically generated

Figure : :Sample Data for Patient

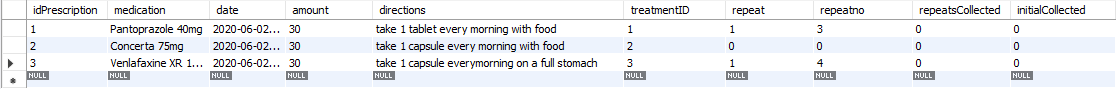


Figure : :Sample Data for Prescription

A screenshot of a social media post

Description automatically generated

Figure : :Sample Data for SystemCall

A screenshot of a cell phone

Description automatically generated

Figure :Sample Data for SystemCallPatient

A screenshot of a cell phone

Description automatically generated

Figure : Sample Data for Test

A screenshot of a cell phone

Description automatically generated

Figure :Sample Data for Treatment

A screenshot of a social media post

Description automatically generated

Figure :Sample Data for Visit

## User Views

A screenshot of a social media post

Description automatically generated

Figure : SQL Code to display which patients require a follow up from a CHW



Figure : Output for Follow-up View

A screenshot of a map

Description automatically generated

Figure : Code To Display Which Precriptions Belong To Which Patients By Their ID numbers (Avila, 2014)

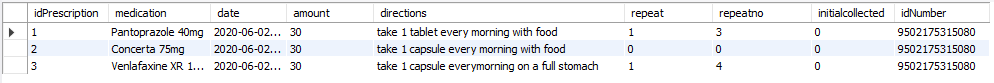


Figure : Output of Prescription View

## Security Mechanisms

A picture containing bird

Description automatically generated

Figure : User list in MySQL (MySQL :: MySQL 8.0 Reference Manual :: 6.1.3 Making MySQL Secure Against Attackers, 2020)

A screenshot of a cell phone

Description automatically generated

Figure : MySQL Login Request (MySQL :: MySQL 8.0 Reference Manual :: 6.1.3 Making MySQL Secure Against Attackers, 2020)

A screenshot of a social media post

Description automatically generated

Figure : Root Password Settings (MySQL :: MySQL 8.0 Reference Manual :: 6.1.3 Making MySQL Secure Against Attackers, 2020)

A screenshot of a cell phone

Description automatically generated

Figure : Password Settings for Community Healthcare Worker (MySQL :: MySQL 8.0 Reference Manual :: 6.1.3 Making MySQL Secure Against Attackers, 2020)

A screenshot of a cell phone

Description automatically generated

Figure : Password Settings for Clinician (MySQL :: MySQL 8.0 Reference Manual :: 6.1.3 Making MySQL Secure Against Attackers, 2020)

A screenshot of a cell phone

Description automatically generated

Figure : Password Settings for Pharmacist (MySQL :: MySQL 8.0 Reference Manual :: 6.1.3 Making MySQL Secure Against Attackers, 2020)

A screenshot of a cell phone

Description automatically generated

Figure : Password Settings for Receptionist (MySQL :: MySQL 8.0 Reference Manual :: 6.1.3 Making MySQL Secure Against Attackers, 2020)

A screenshot of a social media post

Description automatically generated

Figure : Permission Grant Statements For All Above Users (MySQL :: MySQL 8.0 Reference Manual :: 6.1.3 Making MySQL Secure Against Attackers, 2020) (MySQL: Grant/Revoke Privileges, n.d.)

# Deliverable 4: Advanced Query Processing

## Cursor

A screenshot of a cell phone

Description automatically generated

Figure : Code to Create a Cursor Which Accesses Query Results One-By-One (Gravelle, 2013) (MySQL :: MySQL 8.0 Reference Manual :: B.4.6.2 TEMPORARY Table Problems, 2020) (Working with Cursors : MySQL, 2020)

A screenshot of a cell phone

Description automatically generated

Figure : Outputs for the Cursor (Note the Multiple Results Tab)

## Stored Procedure

A screenshot of a social media post

Description automatically generated

Figure :Stored Procedure Code

A screenshot of a cell phone

Description automatically generated

Figure :Stored Procedure Output

## Function

A screenshot of a social media post

Description automatically generated

Figure :Stored Function Code (Black, 2011) (Martinez, 2012) (MySQL :: MySQL 5.7 Reference Manual :: 23.7 Stored Program Binary Logging, 2020) (MySQL :: MySQL 8.0 Reference Manual :: 13.7.4.1 CREATE FUNCTION Syntax for User-Defined Functions, 2020)

A screenshot of a cell phone

Description automatically generated

Figure :Stored Function Output

## Triggers

A screenshot of a social media post

Description automatically generated

Figure : Triggers on the Patient Table to Ensure Valid Information (Razzoli, 2012)

A screenshot of a social media post

Description automatically generated

Figure : Trigger Forcing an Error Due to an Invalid Email

A screenshot of a social media post

Description automatically generated

Figure : Trigger Forcing an Error Due to an Invalid Phone Number

A screenshot of text

Description automatically generated

Figure : Trigger to Ensure Parent Entity Exists for Older Versions of MySQL (Razzoli, 2012)

A screenshot of a social media post

Description automatically generated

Figure :Trigger to Ensure Parent Entity Exists for Older Versions of MySQL (Razzoli, 2012)

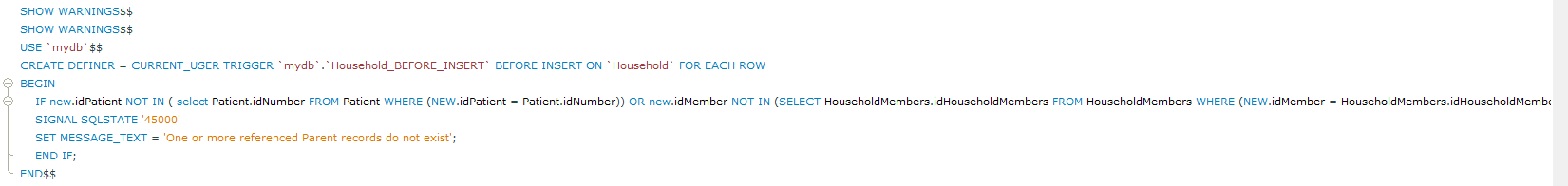


Figure :Trigger to Ensure Parent Entity Exists for Older Versions of MySQL (Razzoli, 2012)

A screenshot of a survey

Description automatically generated

Figure :Trigger to Ensure Parent Entity Exists for Older Versions of MySQL (Razzoli, 2012)

# Deliverable 5: Database Interface Program

# Interface

A screenshot of a social media post

Description automatically generated

Figure : Initial Interface on Load. Displays Whether the Database Managed to Connect.

A screenshot of a social media post

Description automatically generated

Figure : Community Healthcare Worker View After Selecting a CHW Number and Clicking Fetch Data (Dickson, 2011)

A screenshot of a social media post

Description automatically generated

Figure : Prescription View Which is Automatically Refreshed When the Tab is Selected (TabPage Click Events, 2012)

# References

Avila, P., 2014. *Exploring Join Paths; The Key To Building Multiple-Table Joins - Interface Technical Training*. [online] Interface Technical Training. Available at: <https://www.interfacett.com/blogs/exploring-join-paths-the-key-to-building-multiple-table-joins/> [Accessed 26 May 2020].

Black, J., 2011. *Creating Temporary Tables In Mysql Stored Procedure*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/5308969/creating-temporary-tables-in-mysql-stored-procedure> [Accessed 27 May 2020].

Castro, R., 2018. *4 Database Documentation / Data Dictionary Generator Plugins For Mysql Workbench - Mysql Workbench Tutorials*. [online] Dataedo.com. Available at: <https://dataedo.com/kb/tools/mysql-workbench/documentation-generator-plugins> [Accessed 21 April 2020].

Connolly, T. and Begg, C., 2014. *Database Systems: A Practical Approach To Design, Implementation, And Management: Global Edition*. 6th ed. Harlow: Pearson Education, pp.503-583.

Dickson, A., 2011. *Mysql Returns Only One Row*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/4372197/mysql-returns-only-one-row> [Accessed 28 May 2020].

Gravelle, R., 2013. *Mysql Cursors And Loops — Databasejournal.Com*. [online] Databasejournal.com. Available at: <https://www.databasejournal.com/features/mysql/mysql-cursors-and-loops.html> [Accessed 27 May 2020].

Martinez, P., 2012. *Mysql Error When Adding Function*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/11316008/mysql-error-when-adding-function> [Accessed 27 May 2020].

Dev.mysql.com. 2020. *Mysql :: Mysql 5.7 Reference Manual :: 23.7 Stored Program Binary Logging*. [online] Available at: <https://dev.mysql.com/doc/refman/5.7/en/stored-programs-logging.html> [Accessed 27 May 2020].

Dev.mysql.com. 2020. *Mysql :: Mysql 8.0 Reference Manual :: 13.7.4.1 CREATE FUNCTION Syntax For User-Defined Functions*. [online] Available at: <https://dev.mysql.com/doc/refman/8.0/en/create-function-udf.html> [Accessed 27 May 2020].

Dev.mysql.com. 2020. *Mysql :: Mysql 8.0 Reference Manual :: 6.1.3 Making Mysql Secure Against Attackers*. [online] Available at: <https://dev.mysql.com/doc/refman/8.0/en/security-against-attack.html> [Accessed 26 May 2020].

Dev.mysql.com. 2020. *Mysql :: Mysql 8.0 Reference Manual :: B.4.6.2 TEMPORARY Table Problems*. [online] Available at: <https://dev.mysql.com/doc/refman/8.0/en/temporary-table-problems.html> [Accessed 27 May 2020].

Techonthenet.com. n.d. *Mysql: Grant/Revoke Privileges*. [online] Available at: <https://www.techonthenet.com/mysql/grant\_revoke.php> [Accessed 26 May 2020].

Razzoli, F., 2012. *Validating Data Using A TRIGGER*. [online] FALSE IS NOT NULL. Available at: <https://falseisnotnull.wordpress.com/2012/11/06/validating-data-using-a-trigger/> [Accessed 23 May 2020].

Stack Overflow. 2012. *Tabpage Click Events*. [online] Available at: <https://stackoverflow.com/questions/9684221/tabpage-click-events> [Accessed 28 May 2020].

Brainbell.com. 2020. *Working With Cursors : Mysql*. [online] Available at: <https://www.brainbell.com/tutorials/MySQL/Working\_With\_Cursors.htm> [Accessed 26 May 2020].