**CS 5600 Group Assignment: Database Description (20 Points)**

The database description and business rules are very important for design the database system. Your group should work together to create the database description and design a draft of NoSQL Database model. The submission includes with

* Database description.
* A NoSQL Database model.
* Sample data.

**Note.**

You can submit this group assignment in the document files such as .doc, .docx, .pdf, …, etc. and upload on the Black Board.

**For example: CS5600DESIGN\_XXXXX\_YYYYY.doc, where XXXXX is CRN and YYYYY is your group name.**

***Sample***

**Database Description and Business Rules (10 Points)**

The movies are rented out in stores and there are several stores. Each store has a unique distributor that supplies the store with disks. A distributor may supply more than one store. Each distributor has a name, an address, and a phone number. Each store has a name, an address, and a phone number. For each employee we must keep the following information: working store, a name, a supervisor, an address, a phone number, SSN (social security number). For each customer we keep the following information: a name, an address, and a phone number.

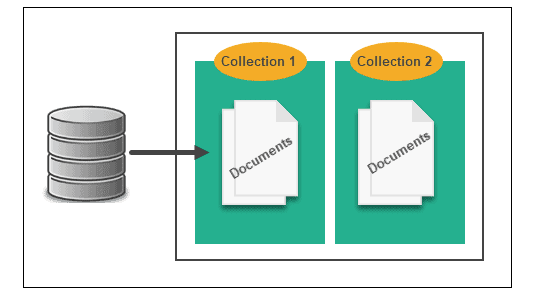
For each rental, we must keep track of which employee served the customer, which movie and which copy (i.e. disk) the customer rented, information about payment, the date and time of the rental, the status (rented, returned in time, returned late), the rate (i.e. the price), and if applicable, due date and overdue charges. About the payment we keep which of the employees accepted the payment (does not have to be the same employee who rented the disk), the type of payment (i.e. cash, check, credit card, direct debit), the amount of the payment, date and time of the payment, payment status

About each disk, we keep information in what condition the disk is and what movie is on the disk. About each movie we keep its title, director’s name, simple description, the name of a (single) major star, the movie’s rating.

**NoSQL Database Model (10 Points)**

**Document Database Model**

**Document-Based Store(JSON File): All data is stored in a document, so there’s no need for cross-referencing and instead of storing information in a table, it’s stored in a document.**

****

**Sample: Amazon Earphones Review**

**Collection 1- Product Info contains:**

**MRP**

**Price**

**Product Full Name**

**Product Short Name**

**Id**

**Collection 2- All Product Reviews contains:**

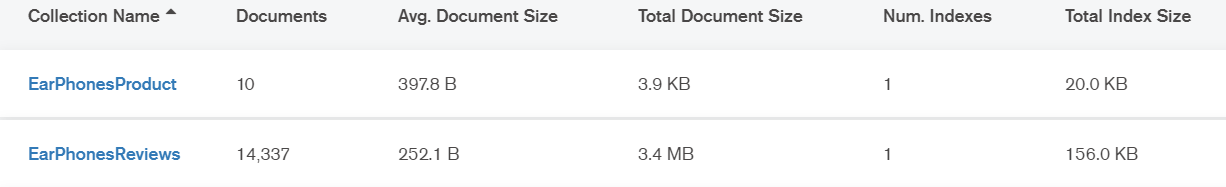
**Review Title**

**Review Body**

**Review Star (Rating)**

**Product (Product Short Name)**

**Sample data: Amazon Earphones Review**

****

**EarPhonesProduct:**

****

**EarPhonesReviews:**

****

**Sample: Order Product**

Product

Prod\_id

Prod\_name

Prod\_stock

Prod\_reorder

Prod\_desc

Prod\_price

Prod\_cost

Order

Ord\_id

Ord\_date

Ord\_amount

Ord\_payment

Cust\_name

Items[

Prod\_name

Qty

]

Customer

Cust\_id

Cust\_name

Cust\_address

Cust\_phone

**Graph Database Model (1)**

**A picture containing text, clipart

Description automatically generated**

**Users publishing posts (:User)-[:PUBLISHED]->(:Post)**

**Tags tagging posts (:Tag)-[:TAGGED]->(:Post)**

**Users liking posts (:User)-[:LIKED]->(:Post)**

**Nodes: User, Tag, and Post.**

**Relationships: PUBLISED, TAGGED, and LIKED.**

**Properties:**

**User (user id, user name, password)**

**Post (post id, title, content, tags, number of like)**

**Tag (tag id, tag name)**

**PUBLISHED (publish date)**

**LIKED (like date)**

**Graph Database Model (2)**

**Diagram

Description automatically generated**

**User loves user (:User)-[:LOVES]->(:User)**

**User dislikes user (:User)-[:DISLIKES]->(:User)**

**User friends of user (:User)-[:FRIENDS\_OF]->(:User)**

**…**

**Nodes: User**

**Relationships: LOVES, DISLIKES, FRIEND\_OF, MARRIED\_TO, COLLEAGUE\_OF, ….**

**Properties: ….**