To: Replica Toys
From: Saharath Kleips, Benjamin Nicholes, Yuta Miyake
Date: 04/14/16
Subject: Project Part 3

Data Model

Our database design has taken into consideration the potential post sales needs of Replica Toys. Throughout the design we considered what would be the easiest way to provide such information such as:

- Product Registration
- Post Sales Surveys
- Product Reviews
- Product Issues/Complaints
- Returns

Here is a brief summarization of the entities in this database that allow us to accomplish this. The physical and logical designs are identical.

- Registration, Registration Response, Answer, Questions – These entities store and map a customer's response to registration questions that are answered upon purchase. The questions and answers are mapped to a specific question number.
- Survey, Survey Response, Answer, Questions – These entities store and map a customer's response to survey questions that are conducted yearly by Replica Toys. The questions and answers are mapped to a specific question number.
- Toy – The toy entity has specific information on a specific toy.
- Model – A line of toys
- Problem Report, Problem Type, Complaint Type, Other Reporter, Test Report, Return Entry, Return Type – These entities handle an issue if one arises with a customer about a toy. Replica Toys will be able to see where a problem came from, what kind of problem occurred, and results of internal testing for issues.
- Review, Review Source – These entities will allow Replica Toys to keep track of external reviews on their products and track where they came from.
- Employee, Distributor, Customer, Zip – Store basic information for people that have purchased, distributed, or are responding to a request.

Processes Needed

Although our database is simple and efficient there are a few things to note:

- It was implemented using the SQL Server Database Administration System
- It is suggested to design and utilize a GUI for ease of use within the business. Allowing people who are not familiar with a language like SQL to easily produce queries.

- Views should be considered when implementing additional features as this can abstract away from complexity as well as providing a security feature.
- Consider the bare metal that the SQL server will be run on, especially the amount of memory it is running and the disk the information is kept on. Speed of processing large amounts of data are dependent on the hardware of the machine.
- Connecting existing datasets should be something to consider. If you want to import bulk existing data there are ways to do this through Excel, SQL Server and other database management systems.

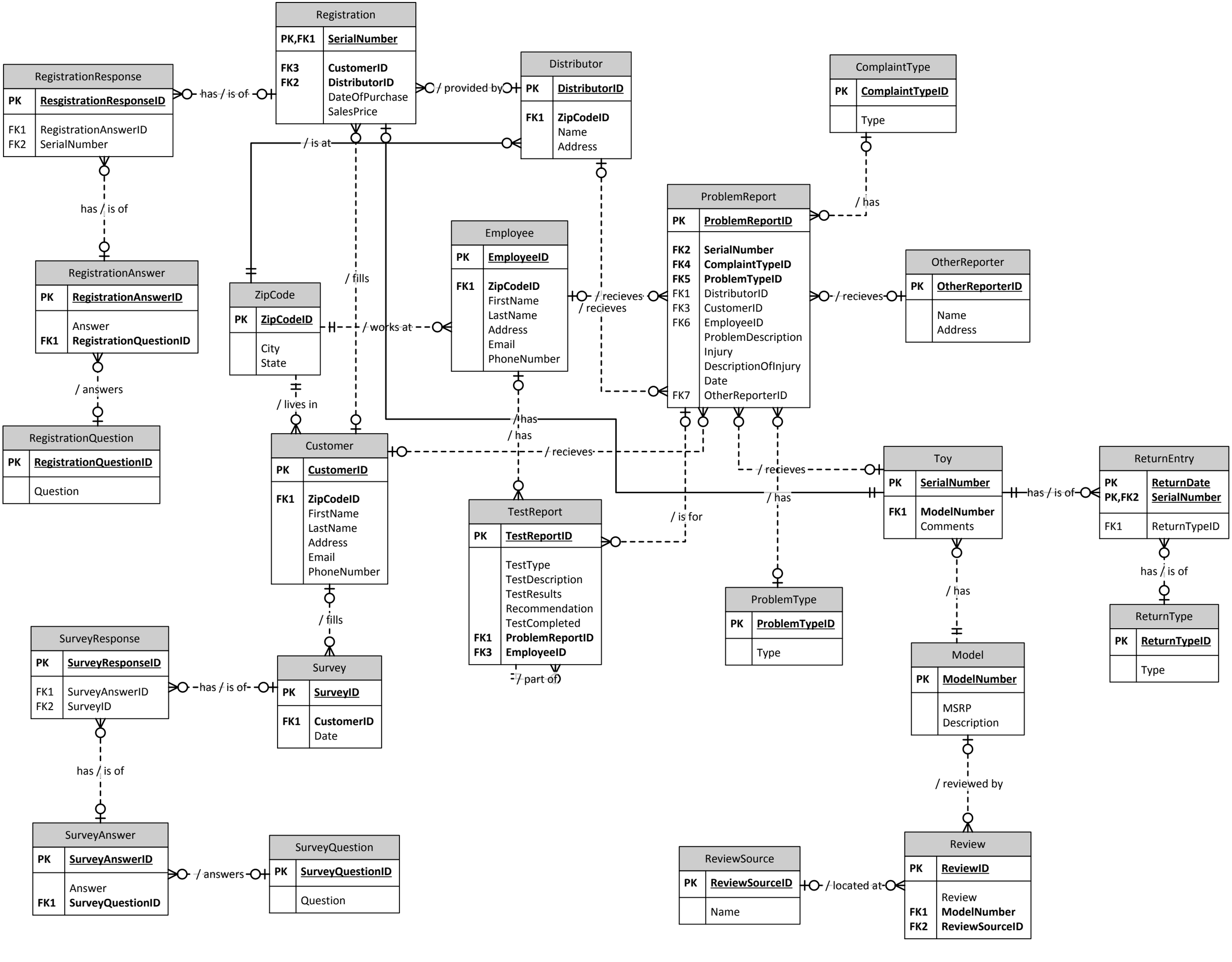
Next Steps

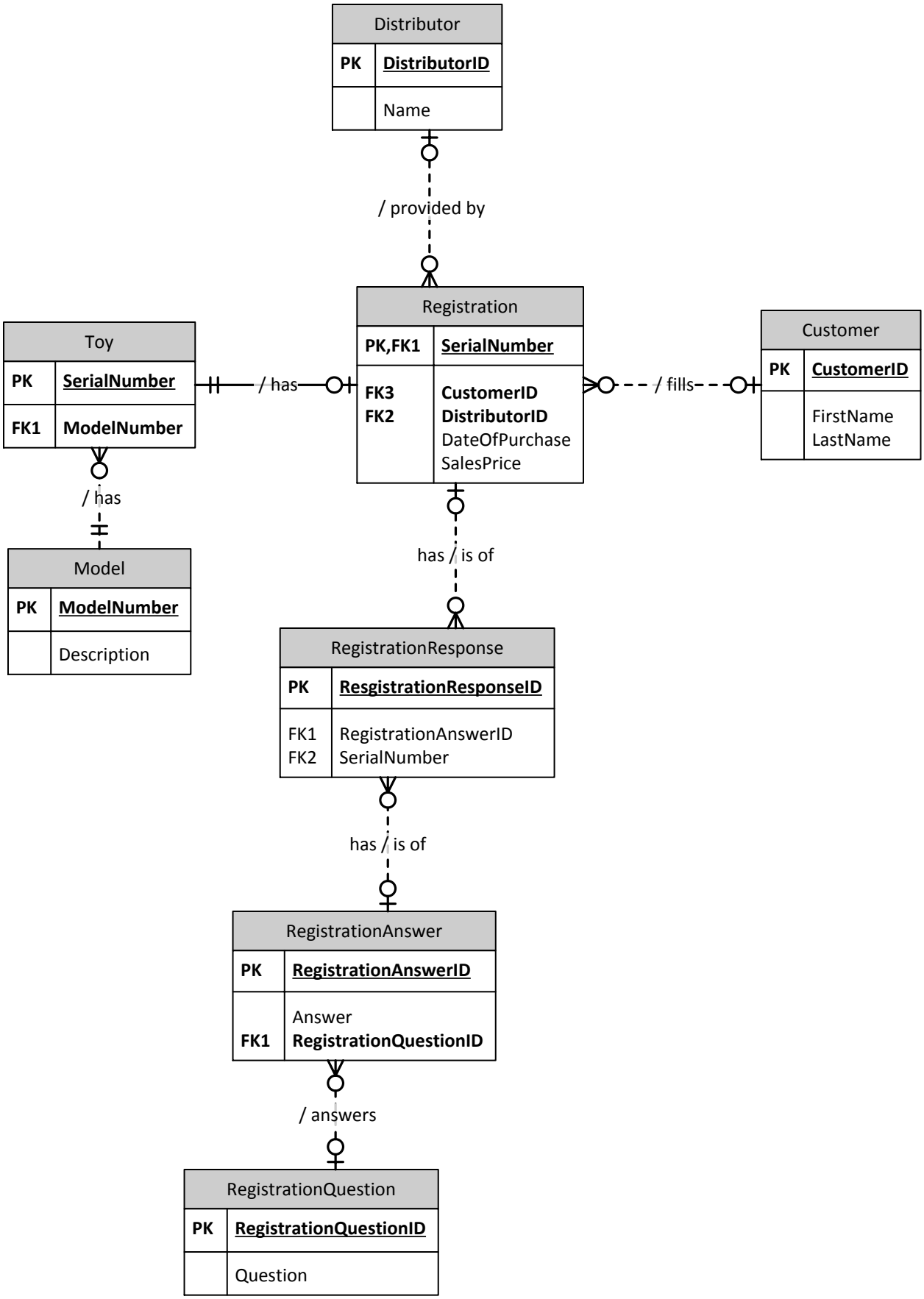
Through the creation of this database we have completed the Planning, Analysis, and Design phases of the Systems Development Life Cycle. A working database implementation has been created. Now the next steps are

- Implementation
    - Creating all necessary documentation for either our group to continue designing additional features or for another engineering team to take over to implement things like a GUI, Data pipelining process, and Server Purchase/Setup.
    - Creating steps and procedures for the business team and customer support to utilize our system from a non-SQL perspective. This would include hands-on training with your staff.
- Maintenance
    - Our group is available per contract of continued maintenance of this database implementation. If there is anything certain in technology, it will be that someday, something will not work. We are ready to handle this and our talented engineering team will jump and provide assistance wherever it is needed, whether this is data integrity issues, performance issues, or any sort of recovery.

Our contracted database is now complete and will be handed over to you with SQL query examples for probable business needs. Please let us know if this meets your standards. If it does not we would be happy to go over them in great detail, and if need start again in the design process.

Database is located in skleips
Comments for views are in source code.


Registration
PK,FK1 SerialNumber
FK3 CustomerID
FK2 DistributorID
DateOfPurchase
SalesPrice
RegistrationResponse
PK ResgistrationResponseID
FK1 RegistrationAnswerID
FK2 SerialNumber
has / is of
has / is of
Distributor
PK DistributorID
FK1 ZipCodeID
Name
Address
/ provided by
/ is at
ComplaintType
PK ComplaintTypeID
Type
/ has
RegistrationAnswer
PK RegistrationAnswerID
Answer
FK1 RegistrationQuestionID
/ answers
ZipCode
PK ZipCodeID
City
State
/ fills
/ works at
/ lives in
Employee
PK EmployeeID
FK1 ZipCodeID
FirstName
LastName
Address
Email
PhoneNumber
/ recieves
/ recieves
ProblemReport
PK ProblemReportID
FK2 SerialNumber
FK4 ComplaintTypeID
FK5 ProblemTypeID
FK1 DistributorID
FK3 CustomerID
FK6 EmployeeID
ProblemDescription
Injury
DescriptionOfInjury
Date
FK7 OtherReporterID
/ recieves
OtherReporter
PK OtherReporterID
Name
Address
RegistrationQuestion
PK RegistrationQuestionID
Question
Customer
PK CustomerID
FK1 ZipCodeID
FirstName
LastName
Address
Email
PhoneNumber
/ has
/ has
/ recieves
/ recieves
/ has
Toy
PK SerialNumber
FK1 ModelNumber
Comments
has / is of
ReturnEntry
PK,FK2 ReturnDate
PK,FK2 SerialNumber
FK1 ReturnTypeID
has / is of
ReturnType
PK ReturnTypeID
Type
TestReport
PK TestReportID
TestType
TestDescription
TestResults
Recommendation
TestCompleted
FK1 ProblemReportID
FK3 EmployeeID
/ is for
/ part of
ProblemType
PK ProblemTypeID
Type
/ has
/ fills
SurveyResponse
PK SurveyResponseID
FK1 SurveyAnswerID
FK2 SurveyID
has / is of
Survey
PK SurveyID
FK1 CustomerID
Date
has / is of
Model
PK ModelNumber
MSRP
Description
/ reviewed by
SurveyAnswer
PK SurveyAnswerID
Answer
FK1 SurveyQuestionID
/ answers
SurveyQuestion
PK SurveyQuestionID
Question
ReviewSource
PK ReviewSourceID
Name
/ located at
Review
PK ReviewID
Review
FK1 ModelNumber
FK2 ReviewSourceID


Distributor
PK
DistributorID
Name
/ provided by
Registration
PK,FK1
SerialNumber
FK3
FK2
CustomerID
DistributorID
DateOfPurchase
SalesPrice
Toy
PK
SerialNumber
FK1
ModelNumber
/ has
/ has
Customer
PK
CustomerID
FirstName
LastName
/ fills
Model
PK
ModelNumber
Description
has / is of
RegistrationResponse
PK
ResgistrationResponseID
FK1
FK2
RegistrationAnswerID
SerialNumber
has / is of
RegistrationAnswer
PK
RegistrationAnswerID
Answer
FK1
RegistrationQuestionID
/ answers
RegistrationQuestion
PK
RegistrationQuestionID
Question

```
-- VIEWS --
-- View for Problem 2 & 3, represents the abstraction of a summary of registrations, which
fully answers problem 2 and is necessary for reuse in problem 3.
CREATE VIEW registrationSummary AS
SELECT RegistrationAnswer.Answer RelationshiptoUser,
       COUNT(*) CountofRegistrations,
       COUNT(DISTINCT Toy.ModelNumber) CountofDistinctModels,
       SUM(Registration.SalesPrice) SumofPrice,
       AVG(Registration.SalesPrice) AveragePrice
FROM Registration
INNER JOIN Toy
ON Registration.SerialNumber = Toy.SerialNumber
INNER JOIN RegistrationResponse
ON Registration.SerialNumber = RegistrationResponse.SerialNumber
INNER JOIN RegistrationAnswer
ON RegistrationResponse.RegistrationAnswerID = RegistrationAnswer.RegistrationAnswerID
GROUP BY RegistrationAnswer.RegistrationAnswerID,RegistrationAnswer.Answer,
RegistrationAnswer.RegistrationQuestionID
HAVING  RegistrationAnswer.RegistrationQuestionID = '21';

-- View for Problem 5, represents all models owned specifically by grandparents, used to find
top 1
CREATE VIEW ModelOwnedByGrandParents AS
SELECT Toy.ModelNumber ModelPurchasedMostbyGrandparents,
       COUNT(*) CountOfModels
FROM Registration
INNER JOIN Toy
ON Registration.SerialNumber = Toy.SerialNumber
INNER JOIN RegistrationResponse
ON Registration.SerialNumber = RegistrationResponse.SerialNumber
INNER JOIN RegistrationAnswer
ON RegistrationResponse.RegistrationAnswerID = RegistrationAnswer.RegistrationAnswerID
GROUP BY Toy.ModelNumber, RegistrationAnswer.RegistrationAnswerID
HAVING RegistrationAnswer.RegistrationAnswerID = '32';

-- View for Problem 7, represents all registration data that belongs to a grandparent
CREATE VIEW GrandParents AS
SELECT Registration.serialNumber
FROM Registration
INNER JOIN Toy
ON Registration.SerialNumber = Toy.SerialNumber
INNER JOIN RegistrationResponse
ON Registration.SerialNumber = RegistrationResponse.SerialNumber
INNER JOIN RegistrationAnswer
ON RegistrationResponse.RegistrationAnswerID = RegistrationAnswer.RegistrationAnswerID
WHERE RegistrationAnswer.RegistrationAnswerID = '32'

-- View for Problem 8, represents the feature response of grandparents
CREATE VIEW GrandParentCareFeatures AS
SELECT  Answer "Feature",
        count(*) "Count of Feature mentioned"
FROM RegistrationResponse
INNER JOIN RegistrationAnswer
ON RegistrationResponse.RegistrationAnswerID = RegistrationAnswer.RegistrationAnswerID
WHERE RegistrationAnswer.RegistrationQuestionID = '83'
      AND RegistrationResponse.serialNumber in (SELECT serialNumber FROM GrandParents)
GROUP BY answer
```

```sql
-- PROBLEM 1
SELECT  Registration.SerialNumber "Serial Number",
        Toy.ModelNumber "Model Number",
        Model.Description "Model Description",
        Customer.LastName + ', ' + Customer.FirstName "Buyer Name",
        CONVERT(VARCHAR(10), Registration.DateOfPurchase, 101) "Purchase Date",
        Registration.SalesPrice "Price",
        Distributor.Name "Distributor",
        RegistrationAnswer.Answer "Relationship of Buyer to User"
FROM Registration
INNER JOIN Toy
ON Registration.SerialNumber = Toy.SerialNumber
INNER JOIN Model
ON Toy.ModelNumber = Model.ModelNumber
INNER JOIN Customer
ON Registration.CustomerID =  Customer.CustomerID
INNER JOIN Distributor
ON Registration.DistributorID = Distributor.DistributorID
INNER JOIN RegistrationResponse
ON Registration.SerialNumber = RegistrationResponse.SerialNumber
INNER JOIN RegistrationAnswer
ON RegistrationResponse.RegistrationAnswerID = RegistrationAnswer.RegistrationAnswerID
WHERE RegistrationAnswer.RegistrationQuestionID = '21';
```

	Serial Number	Model Number	Model Description	Buyer Name	Purchase Date	Price	Distributor	Relationship of Buyer to User
1	72544274	88361412	Pagani Huayra	Lagares, Juan	04/05/2015	652.00	Cars R Us	Grandparent
2	46714293	83985841	Zenvo ST1	Matz, Steven	07/06/2015	540.00	Snaks	Parent
3	86759623	88408109	McLaren F1	Granderson, Curtis	10/24/2012	582.00	Toy Planet	Parent
4	52377834	24665852	Koenigsegg CCX	Syndergaard, Noah	08/03/2015	176.00	Convertables and Co.	Family Friend
5	50488423	69667517	Saleen S7 TT	Cabrera, Asdrubal	09/07/2015	327.00	Snaks	Parent
6	82513149	92727404	Bugatti Veyron EB 16.4	Harvey, Matt	06/24/2014	273.00	Convertables and Co.	Parent
7	42464912	16611522	McLaren 650S	DeGrom, Jacob	01/03/2014	254.00	Little Wheels	Family Friend
8	80830234	83496832	SSC Ultimate Aero	Verrett, Logan	06/25/2014	358.00	Snaks	Parent
9	21847715	91595048	9ff GT9-R Porsche	Collins, Terry	06/26/2014	185.00	Convertables and Co.	Parent
10	86928760	52431160	Koenigsegg Agera R	Familia, Jeurys	06/27/2014	126.00	Convertables and Co.	Grandparent
11	11293419	29646249	Koenigsegg One:1	Plawecki, Kevin	03/03/2013	110.00	Toy Planet	Other
12	24896934	61566272	Bugatti Veyron Super Sport	Blevins, Jerry	01/02/2016	587.00	Convertables and Co.	Other
13	11111111	73295891	Hennessey Venom GT	Blevins, Jerry	01/30/2014	345.00	Toy Planet	Grandparent
14	22222222	73295891	Hennessey Venom GT	Plawecki, Kevin	07/07/2012	550.00	Cars R Us	Grandparent
15	33333333	73295891	Hennessey Venom GT	Familia, Jeurys	08/03/2015	286.00	Convertables and Co.	Grandparent
16	44444444	61566272	Bugatti Veyron Super Sport	Blevins, Jerry	06/24/2014	300.00	Cars R Us	Grandparent
17	92423536	84771879	Porsche 918 Spyder	Wright, David	01/30/2014	141.00	Big Wheels	Family Friend
18	18567525	96821953	McLaren P1	Walker, Neil	09/02/2012	486.00	Toy Planet	Grandparent
19	42675336	61761059	Lamborghini Huracan	Cespedes, Yoenis	12/03/2012	159.00	Vehicles Ltd.	Parent
20	37832480	13903212	Ferrari LaFerrari	Colon, Bartolo	09/02/2012	671.00	Flappy Planet	Grandparent
21	24784636	26533784	Lamborghini Veneno	Duda, Lucas	05/01/2011	233.00	Snaks	Grandparent
22	34463671	28121486	Gumpert Apollo	Flores, Wilmer	02/08/2010	103.00	Hooray Inc	Other
23	14133677	96301188	Noble M600	Conforto, Michael	02/15/2010	434.00	Convertables and Co.	Parent

```sql
-- PROBLEM 2
SELECT  RelationshiptoUser "Relationship to User",
        CountofRegistrations "Count of Registrations",
        CountofDistinctModels "Count of Distinct Models",
        SumofPrice "Sum of Price",
        AveragePrice "Average Price"
FROM registrationSummary;
```

	Relationship to User	Count of Registrations	Count of Distinct Models	Sum of Price	Average Price
1	Parent	8	8	2858.00	357.25
2	Grandparent	9	7	3649.00	405.4444
3	Family Friend	3	3	571.00	190.3333
4	Other	3	3	800.00	266.6666

```sql
-- PROBLEM 3
SELECT  RelationshiptoUser "Relationship to User",
        CountofRegistrations "Count of Registrations",
        CountofDistinctModels "Count of Distinct Models",
        SumofPrice "Sum of Price",
        AveragePrice "Average Price"
FROM registrationSummary
WHERE CountofRegistrations = (  SELECT MAX(CountofRegistrations)
                                FROM registrationSummary);
```

	Relationship to User	Count of Registrations	Count of Distinct Models	Sum of Price	Average Price
1	Grandparent	9	7	3649.00	405.4444

```sql
-- PROBLEM 4
SELECT Toy.ModelNumber "Model Number",
       COUNT(*) "Number of Toys",
       SUM(Registration.SalesPrice) "Sum of Price",
       AVG(Registration.SalesPrice) "Average Price",
       CONVERT(VARCHAR(10), MIN(Registration.DateOfPurchase), 101) "Earliest Registration Date",
       CONVERT(VARCHAR(10), MAX(Registration.DateOfPurchase), 101) "Latest Registration Date"
FROM Registration
INNER JOIN Toy
ON Registration.SerialNumber = Toy.SerialNumber
GROUP BY Toy.ModelNumber;
```

	Model Number	Number of Toys	Sum of Price	Average Price	Earliest Registration Date	Latest Registration Date
1	13903212	1	671.00	671.00	09/02/2012	09/02/2012
2	16611522	1	254.00	254.00	01/03/2014	01/03/2014
3	24665852	1	176.00	176.00	08/03/2015	08/03/2015
4	26533784	1	233.00	233.00	05/01/2011	05/01/2011
5	28121486	1	103.00	103.00	02/08/2010	02/08/2010
6	29646249	1	110.00	110.00	03/03/2013	03/03/2013
7	52431160	1	126.00	126.00	06/27/2014	06/27/2014
8	61566272	2	887.00	443.50	06/24/2014	01/02/2016
9	61761059	1	159.00	159.00	12/03/2012	12/03/2012
10	69667517	1	327.00	327.00	09/07/2015	09/07/2015
11	73295891	4	1477.00	369.25	07/07/2012	08/03/2015
12	83496832	1	358.00	358.00	06/25/2014	06/25/2014
13	83985841	1	540.00	540.00	07/06/2015	07/06/2015
14	84771879	1	141.00	141.00	01/30/2014	01/30/2014
15	88361412	1	652.00	652.00	04/05/2015	04/05/2015
16	88408109	1	582.00	582.00	10/24/2012	10/24/2012
17	91595048	1	185.00	185.00	06/26/2014	06/26/2014
18	92727404	1	273.00	273.00	06/24/2014	06/24/2014
19	96301188	1	434.00	434.00	02/15/2010	02/15/2010
20	96821953	1	486.00	486.00	09/02/2012	09/02/2012

```sql
-- PROBLEM 5
SELECT  ModelPurchasedMostbyGrandParents "Model Purchased Most by Grandparents",
        CountOfModels "Count of Models"
FROM ModelOwnedByGrandParents
WHERE CountOfModels = ( SELECT MAX(CountOfModels)
                        FROM ModelOwnedByGrandParents);
```

	Model Purchased Most by Grandparents	Count of Models
1	73295891	3

```sql
-- PROBLEM 6
SELECT  RegistrationAnswer.Answer "Feature",
        COUNT(*) "Count of Times Feature is Mentioned",
        CAST(COUNT(*)/CAST((SELECT COUNT(DISTINCT SerialNumber) FROM RegistrationResponse) as
        decimal(2)) * 100 AS varchar) + '%' "% of Registrations Mentioned"
FROM RegistrationResponse
RIGHT OUTER JOIN RegistrationAnswer
ON RegistrationResponse.RegistrationAnswerID = RegistrationAnswer.RegistrationAnswerID
GROUP BY RegistrationResponse.RegistrationAnswerID, RegistrationAnswer.RegistrationQuestionID,
RegistrationAnswer.Answer
HAVING RegistrationAnswer.RegistrationQuestionID = '83'
ORDER BY COUNT(*) DESC;
```

	Feature	Count of Times Feature is Mentioned	% of Registrations Mentioned
1	Safety Features	6	26.086900%
2	Color	4	17.391300%
3	Cost	4	17.391300%
4	Sound Features	3	13.043400%
5	Size	2	8.695600%
6	Other	2	8.695600%
7	Quality of Design	1	4.347800%
8	Level of Replication from Original	1	4.347800%
9	Speed	1	4.347800%
10	Type of Toy	1	4.347800%

```sql
-- PROBLEM 7
SELECT  Answer "Feature Mentioned by Grandparents",
        count(*) "Count of Times Feature is Mentioned"
FROM RegistrationResponse
RIGHT OUTER JOIN RegistrationAnswer
ON RegistrationResponse.RegistrationAnswerID = RegistrationAnswer.RegistrationAnswerID
WHERE RegistrationAnswer.RegistrationQuestionID = '83'
      AND RegistrationResponse.serialNumber in (SELECT serialNumber FROM GrandParents)
GROUP BY answer
```

	Feature Mentioned by Grandparents	Count of Times Feature is Mentioned
1	Color	2
2	Cost	2
3	Safety Features	3
4	Size	1
5	Sound Features	1

```
-- PROBLEM 8
SELECT TOP 3 Feature
FROM GrandParentCareFeatures
ORDER BY  "Count of Feature mentioned" DESC;
```


Feature
1
Safety Features
2
Cost
3
Color