

TEAM JURASSIC PARK



Dip & Clip
Grooming Salon and Boarding

Team 3

Robert Nelson

Baden Swallow

Bryan Flynn

Nayan Tez Vankayalapati

Table Of Contents

Executive Summary	2
Team Member Introductions	3
Team Charter and Agreements	4
Member Roles / Expected Schedules	6
Dip and Clip Grooming Salon and Boarding	7
Current Situation and Requirements	8
ERD Progression	9
Logical Relational Schema	12
Database Screen Captures	13
Database Query Screen Captures	25
Visualizations	60
Major Challenges	64
Lessons Learned	65
Appendix A	66
Appendix B	95
Appendix C	104
Appendix D	111

Executive Summary

This report was commissioned to conduct a full comprehensive database construction for Dip & Clip Grooming Salon and Boarding that would provide the business with complete, and up to date, records as well as the ability to offer new services to their customers.

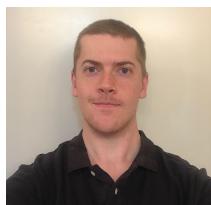
This report shows all stages of the database development process that was used. The first step was to create an accurate and efficient entity-relationship diagram to set the foundation for the database. The database was then created using the tables outlined in the ERD and all past data records were entered into the new database system. Upon finishing the creation of the tables, they were then tested using both basic and advanced tests to ensure the integrity of the tables would not fail. Transactions were then created on the data to show that data could be inserted into the tables without issue and could be removed from the tables just as easily. Finally indexes were added to the tables and the database was tested extensively using queries to ensure that no matter how much data was added the database would hold.

Data Analysis is an integral part of any business maintaining and growing its competitive market in the current market and any future market. With this database Dip & Clip will be able to efficiently manage client transactions like never before and the potential for growth beyond the Flagstaff market will be limitless.

Team Member Introductions



Robert Nelson is a Business Data Analytics major at ASU. He is both a quick start and a follow through which makes him an essential part of our team. He has internship experience at Gates Corporation working with Data Management and Business Intelligence. Going along with Robert being a quick start he is also detail oriented which gives us a key component to success. He has extensive success with data programs such as Tableau and Teradata. Robert can also be found performing in The Light in the Piazza.



Baden Swallow is a Computer Information Systems major here at ASU and he plays an important role in our group as a facilitator and helps to organize our meetings. He has experience in community assistance and also has retail experience with his current job as a delivery driver for Domino's. Baden is methodical in his work and he prefers that we go at a steady pace and get the task done right the first time. Baden often works by himself to complete database systems however he has begun to gain experience in working in groups to complete larger tasks. He is currently looking into gaining education experience in supply chain management.



Bryan Flynn is a business data analytics major here at ASU. He has 3 years of experience in retail doing loss analytics. He would describe himself as a hard worker and a perfectionist. His main role within the group is to be a quick start, he also has a lot of experience with C#, SQL, and excel. He is a Houston, Texas native but plans on moving to Denver, Colorado after college to work in aerospace in data analytics.



Nayan is majoring in Business Data Analytics and Supply Chain Management and is currently a Junior at ASU. Nayan has interned at a Ford wheel supplier company and is familiar with SQL and Microsoft Office Suite. Although relatively new to Database Management, Nayan is looking forward to learning more in detail about databases and Information Systems. Outside of academia, Nayan is a big automobile fan and is looking forward to getting back onto the track!

Team Jurassic Park Charter/Code of Ethics

We, the **Jurassic Park** team, having signed the attached Team Membership Agreement, have developed the following code of ethics to manage our project team activities. This Team Charter/Code specifies the minimum expectations of appropriate behavior for each team member, and will be used to evaluate any potential problems.

If, in the future, we determine that the rules specified in this document should be modified or supplemented, we may produce a revised version. At that time, the revised version will become effective, and the older version will be null.

Code of Conduct: As a project team, we will:

- Treat team members with respect and dignity at all times, regardless of stresses or current temperament.
- Work Diligently to meet deadlines and expectations.
- Complete all work according to the W.P. Carey Honor Code

Participation: We will:

- Each contribute a fair share of the workload assigned to the team and ensure that each individuals work will be tailored towards their particular talent when possible.
- Assign roles according to connotation types so that team members can complete tasks.
- Follow our team guidelines for addressing issues as they arrive.

Communication: We will:

- Bring up any complaints or problems as soon as they arise so that may be dealt with accordingly and to reduce the amount of time spent re-modifying already made work.
- The main form of communication will be through our slack channel, however if it is an urgent matter we will text each other for faster response times.

Problem Solving: We will:

- Notify and address fellow team members of any problems or issues that may occur over the course of this project and deal with said issues as they arise to avoid unnecessary troubles with the project.
- Hold team members accountable to both meeting dates/times and their assigned tasks for the benefit of everyone involved.

Meeting Guidelines: We will:

- Allocate all work for the week at each of our meetings, with the expectation that the work will be completed at each member's earliest possible convenience before the project deadline.

We, the undersigned, accept the conditions of team membership for CIS 365. The specific conditions of team membership are recorded in the Team Charter which we have developed and attached.

The Charter reflects the minimum level of performance that is expected of any group member. If a group member fails to meet this minimum level, they may be subject to the following consequences:

1. Group will discuss reasons why the team is not succeeding. Any individual seen to not be contributing to the group in a meaningful way with no excuse will be assigned a larger portion of work the following week.
2. If we feel that someone is not participating at a necessary level for team success we will take necessary action.
3. Any team member that misses 2 team meetings in a row will be made to go up the stairs of the BA building 5 times before class.

Project Team Name: Jurassic Park

Typed Name:	Signature:	Date:
Baden Swallow	<i>Baden Swallow</i>	09/15/2019
Bryan Flynn	<i>Bryan Flynn</i>	09/15/2019
Robert Nelson	<i>Robert Nelson</i>	09/15/2019
Nayan Tez	<i>Nayan Tez</i>	09/15/2019

Team Profile:

Skills Possessed by Team

● SQL
● Python
● Office Suite
● POS (Point of Sale) Technology
● C#
● Java
● Tableau

Technology Available to Team

● Printer
● Scanner
● Laptops
● Office Suites

Meeting Times:

We meet on Thursdays at 8:00 AM in person along with expectations for online meetings at other times during the week as needed.

Proposed Assigned Tasks:

Baden Swallow: Formation of all Slack & supporting documents required for Final Submission.
Nayan: Final proofing before submission
Robert Nelson: Final Document Formation, Tile Page Development
Byran Flynn: Compile skills, technology, and assigned roles

Dip and Clip Grooming Salon and Boarding



The client that this report is written for is Dip & Clip Grooming Salon and Boarding. Dip & Clip is a small animal grooming salon and boarding kennel located in Flagstaff, Arizona. Dip & Clip owners, Sharon and Jody Prescott, have owned the business for 5 years and have been able to build up a loyal clientele in and around the Flagstaff area. Jody and Sharon have been able to obtain some success and now are interested in streamlining the business' data management processes.

Dip & Clip is currently working with a non computerized, and frankly inefficient, data management system and would like everything to be put in a user friendly database. Dip & Clip currently has 12 full time staff and 4 part time employees and is experiencing issues with new part time employees understanding the confusing data management program. Dip & Clip has tasked Team Jurassic Park with making their business efficient in order to perpetuate future growth.

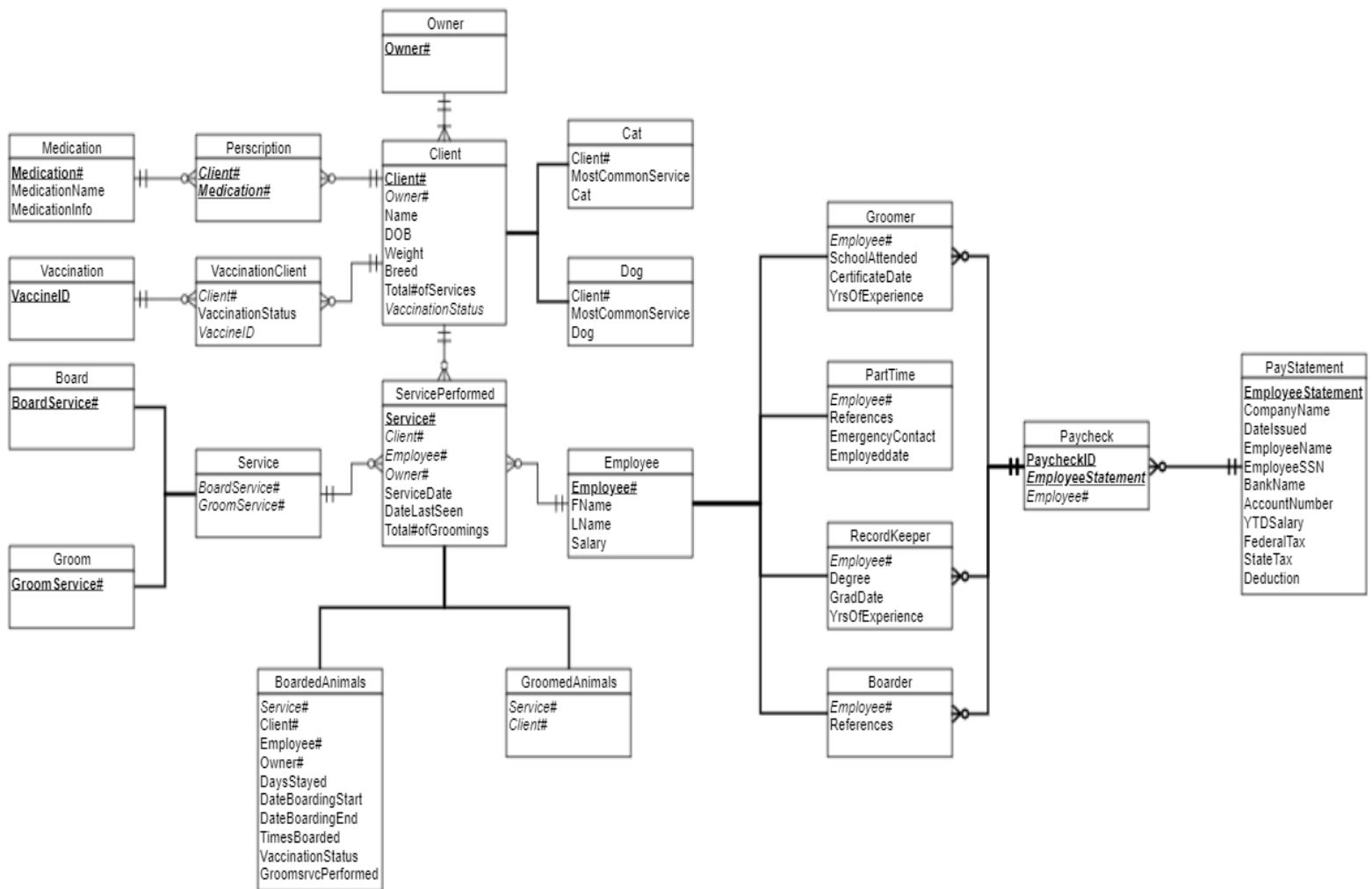
Current Situation and Requirements

Dip and Clip originally stored all of the business' data on paper with no aspect of their information systems being computerized. As the business continues to grow, the complexity of record holding increases. Dip and Clip does not have the ability to track past client information and new employees often struggle with understanding and assimilating in the business model.

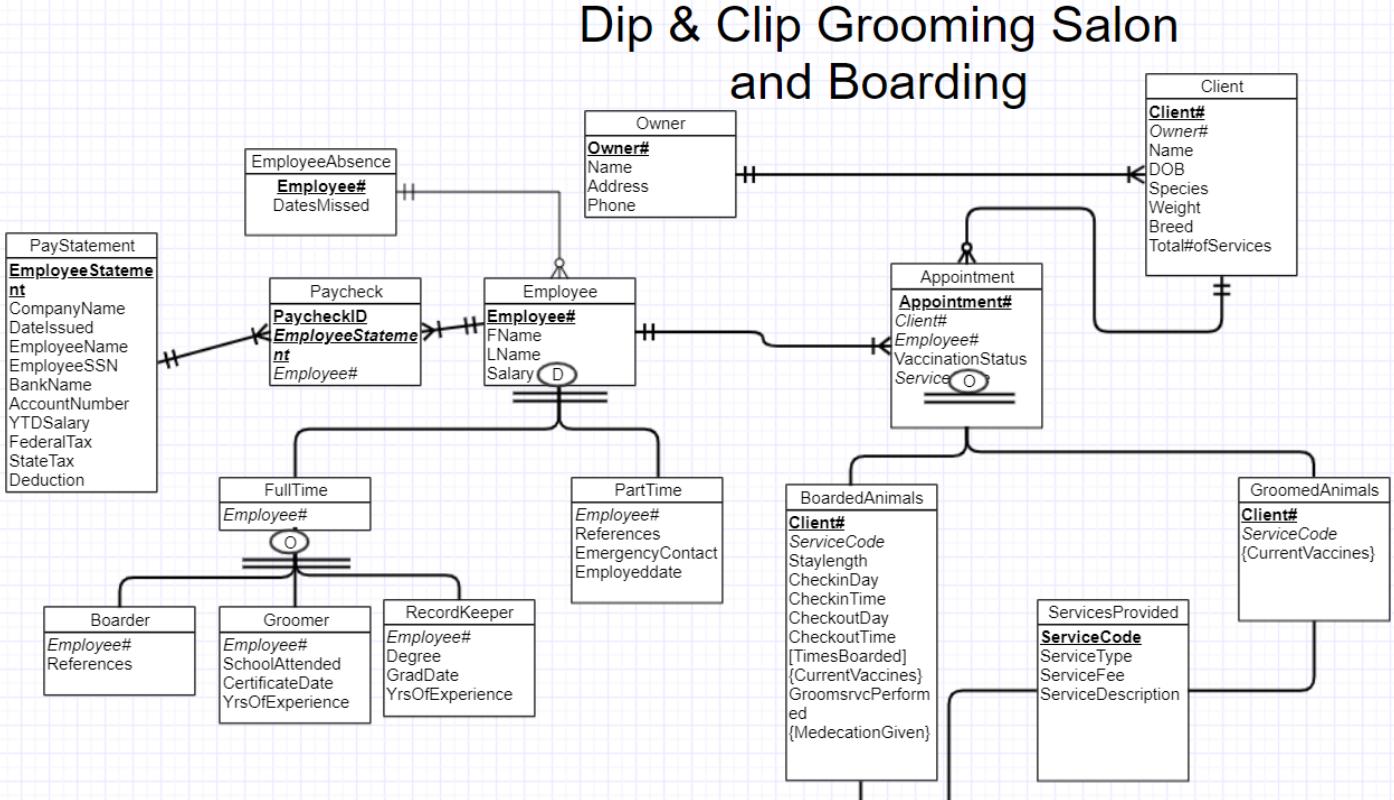
Sharon Prescott, co owner of Dip & Clip, has been advised by her client that she needs to switch from a file information system to a database system using MS SQL Server. Sharon would like all client, owner, appointment, and employee information put into the database and would like for it to serve her for the next 5+ years.

ERD Progression: (Initial Version of ERD)

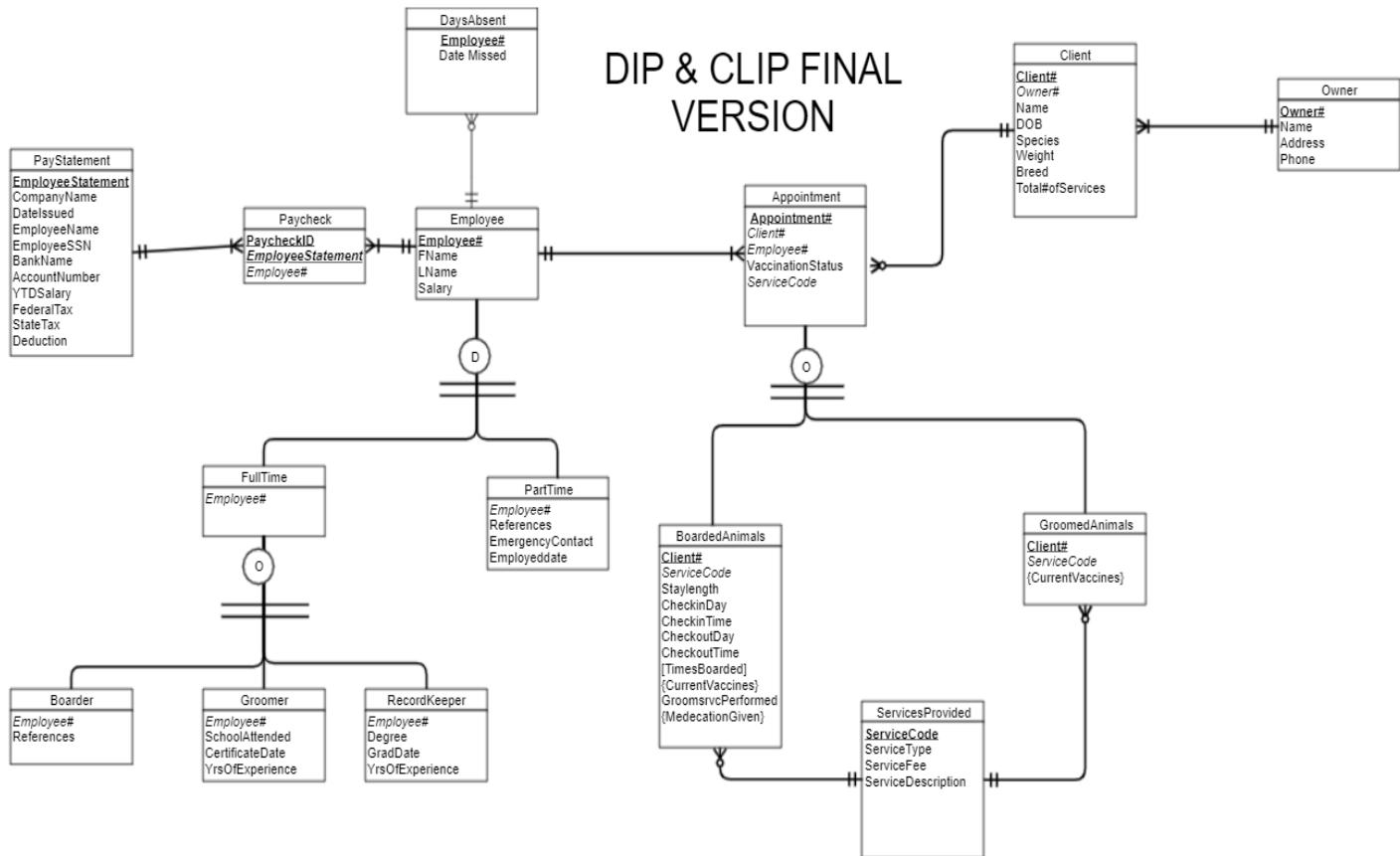
Dip & Clip Grooming Salon and Boarding



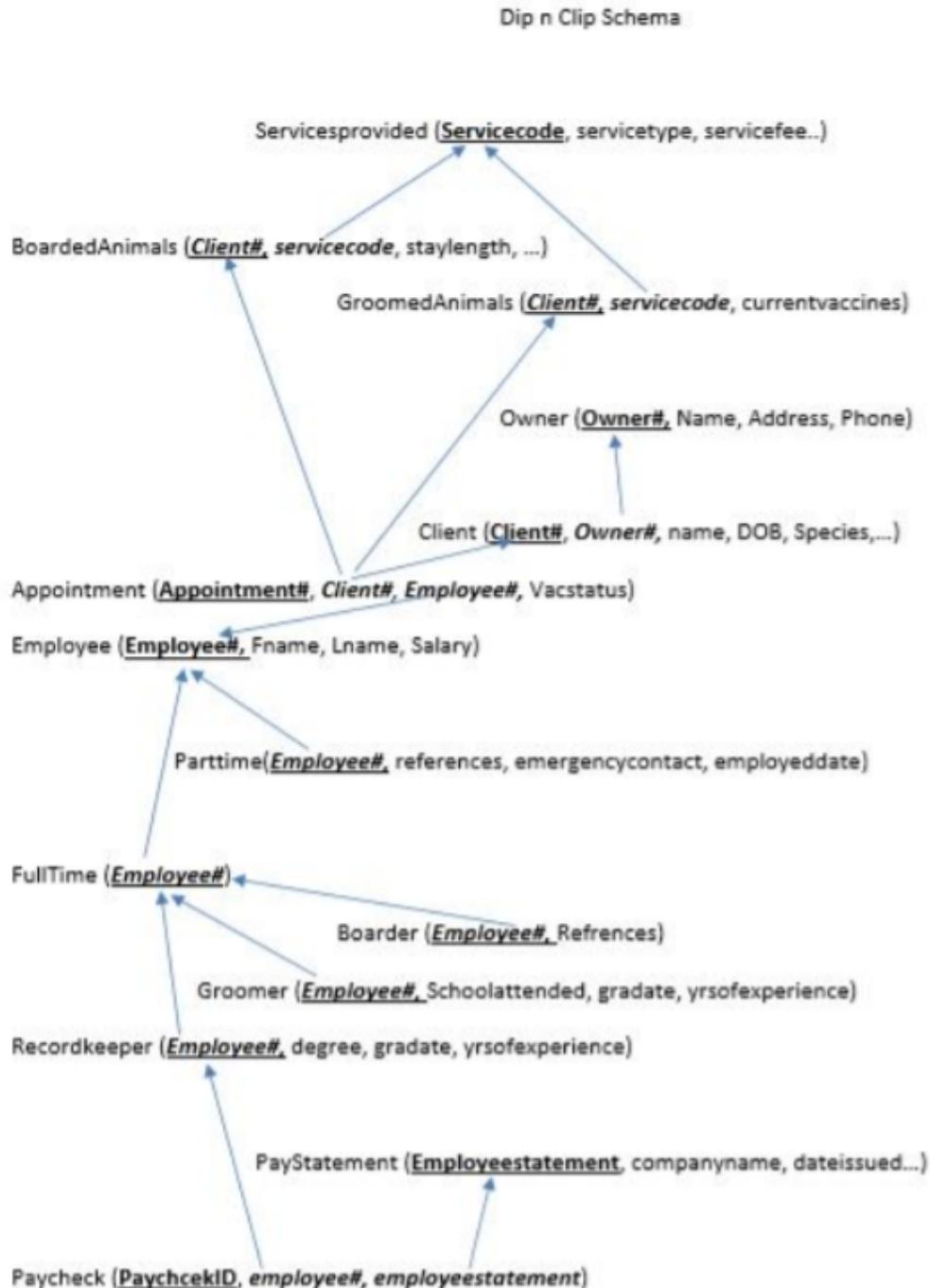
ERD Progression: (Second Version of ERD)



ERD Progression: (Final Updated Version)



Logical Relational Schema



Database Screen Captures

Screenshots of All Tables:

Employee Table:

The screenshot shows a SQL query results window. At the top, a query is written in blue text: "Select * from EMPLOYEE". Below the query, the results are displayed in a table format. The table has 16 rows, each representing an employee with columns: Employee#, EFName, ELName, Salary, and SickDaysUsed. The first few rows of data are as follows:

	Employee#	EFName	ELName	Salary	SickDaysUsed
1	01	Jonathan	Selmy	42450.00	2
2	02	Amanda	Laith	51000.00	1
3	03	Sam	Forwell	46000.00	2
4	04	Riley	Shorwich	23000.00	2
5	05	Dale	Timmel	54000.00	0
6	06	Olivia	Alaway	25000.00	0
7	07	Abigail	Chavez	88000.00	0
8	08	Timothy	Lowell	67000.00	0
9	09	Ross	Anders	32000.00	0
10	10	Grace	Robertson	56000.00	1
11	11	Courtney	Alcine	43000.00	0
12	12	Paul	Elworth	44000.00	0
13	13	Becca	Freed	25000.00	0
14	14	Ellie	Barret	25000.00	0
15	15	Nathan	Meyers	25000.00	1
16	16	David	Andrews	25000.00	0

Client Table:

The screenshot shows a SQL query window and a results grid. The query is:

```
Select *  
from CLIENT
```

The results grid displays 21 rows of data from the CLIENT table, with columns: Client#, Owner#, Name, DOB, Gender, Species, Weight, Breed, and Total#ofServices.

	Client#	Owner#	Name	DOB	Gender	Species	Weight	Breed	Total#ofServices
1	10001	00002	Sparky	2014-03-12	Male	Dog	40	Golden Retriever	1
2	10002	00019	Bubbles	2010-01-02	Male	Dog	22	Basset Hound	3
3	10003	00005	Jazz	2015-11-23	Female	Cat	14	Burmese	2
4	10004	00011	Lucky	2012-11-14	Male	Cat	10	Shorthair	4
5	10005	00001	Toby	2008-06-29	Female	Dog	12	Chinese Crested Dog	1
6	10006	00029	Cali	2017-07-19	Female	Cat	15	Bengal	2
7	10007	00008	Rotor	2014-03-06	Male	Dog	68	Great Dane	2
8	10008	00013	Tyson	2016-07-15	Female	Cat	15	Sphynx	6
9	10009	00007	Scrabs	2015-01-26	Male	Dog	26	Welsh Corgi	5
10	10010	00017	Tiddles	2015-06-11	Female	Dog	38	Chocolate Labrador	3
11	10011	00026	Fabio	2011-09-09	Male	Cat	12	Persian	4
12	10012	00004	Tazzie	2016-02-28	Male	Dog	18	Beagle	1
13	10013	00023	Maggie	2018-02-14	Male	Cat	16	Siamese	2
14	10014	00028	Buttons	2017-05-19	Male	Cat	10	Chartreux	2
15	10015	00030	Chad	2015-07-05	Female	Cat	11	Turkish Van	3
16	10016	00006	Sprit	2017-08-13	Male	Cat	15	Manx	8
17	10017	00014	Clover	2009-12-25	Female	Cat	14	Burmese	6
18	10018	00003	Annabelle	2012-12-07	Female	Dog	22	Jack Russel Terrier	5
19	10019	00025	Lars	2013-06-04	Female	Cat	18	Japanese Bobtail	3
20	10020	00016	Whiskey	2016-04-22	Male	Cat	14	Sphynx	2
21	10021	00009	Rouge	2016-03-09	Male	Dog	46	Golden Retriever	1

Owner Table:

The screenshot shows a SQL query window with the following content:

```
Select *  
from OWNER
```

The results pane displays the data from the OWNER table:

	Owner#	Name	Address	Phone
1	00001	Jack Bauer	54 King Ave.	602-555-0170
2	00002	Garcia Lopez	45 Lakeview Ave.	602-623-8754
3	00003	Trisha Stevens	348 South Tarkiln Hill St.	480-623-4452
4	00004	Alex Tills	386 Buttonwood Ave.	928-623-9781
5	00005	Matthew Alves	83 Park Court Rd.	928-787-6394
6	00006	Zach Martin	383 Littleton Court	928-891-2748
7	00007	Rebecca Roberts	12 Essex St.	602-776-8021
8	00008	Carl Grimes	30 Glenlake Ave.	480-995-2512
9	00009	Daniel Jones	8358 Depot Street	480-341-5659
10	00010	Enrique Vasquez	146 Hillcrest St.	602-415-6362
11	00011	Hannah Barkley	840 E. Roosevelt St.	920-212-2412
12	00012	Evan Jones	22 Rock Creek St.	623-744-6368
13	00013	Julia Roberts	8507 Sleepy Hollow St.	602-542-8567
14	00014	Ian Hamilton	7483 Thomas Circle	602-853-6254
15	00015	Ben Stiller	523 Peachtree Street	480-741-9632
16	00016	Lianne James	21 North Hill Ave.	620-852-7534
17	00017	William Butler	8793 Summit Lane	480-846-2586
18	00018	David Smith	16 Liberty St.	602-165-5971
19	00019	Morgan Hayes	27 N. Vermont Court	623-734-6739
20	00020	Noah Wilson	376 Summerhouse St.	220-563-8336
21	00021	Caleb Jones	5 West Homewood St.	602-443-6631

Full Time Employees:

The screenshot shows a SQL query window and its results. The query is:

```
Select *  
from FULLTIME
```

The results grid displays the following data:

	Employee#
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	08
9	09
10	10
11	11
12	12

Record Keeper Employees:

```
Select *
from RECORDKEEPER
```

150 %

	Employee#	Degree	GradDate	YrsOfExperience
1	01	CIS	2019-12-12	3
2	02	FIN	2018-10-12	4

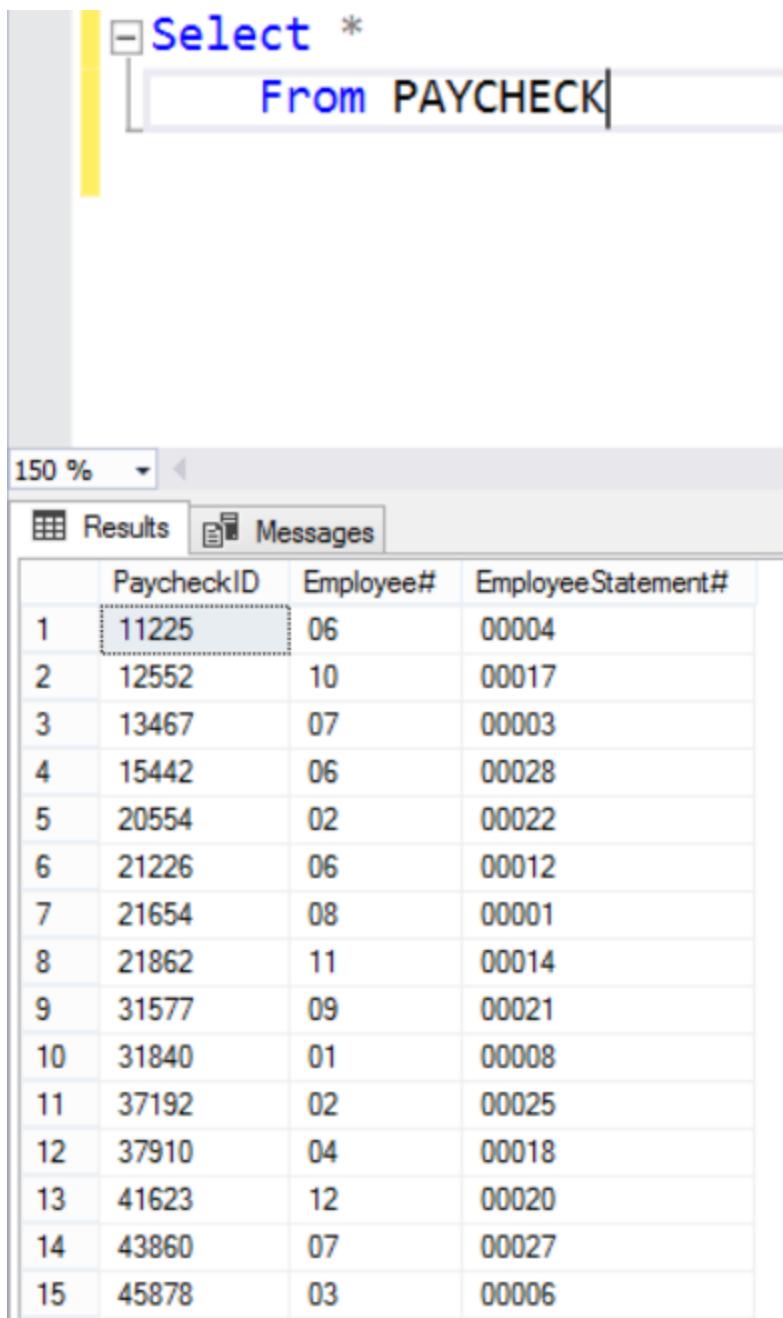
Groomer Employees:

```
Select *
from GROOMER
```

150 %

	Employee#	SchoolAttended	CertificateDate	YrsOfExperience
1	07	ASU	2019-12-12	3
2	08	GCU	2018-10-12	7
3	09	NAU	2019-10-07	2
4	10	NAU	2016-02-01	5
5	11	ASU	2019-01-06	1
6	12	ASU	2019-05-06	3

Paycheck Table:



The screenshot shows a SQL query window and its results grid. The query is:

```
Select *
From PAYCHECK
```

The results grid displays 15 rows of data with columns: PaycheckID, Employee#, and EmployeeStatement#. The PaycheckID column contains values from 1 to 15. The Employee# column contains values from 01 to 12. The EmployeeStatement# column contains values starting with 00004 and ending with 00027.

	PaycheckID	Employee#	EmployeeStatement#
1	11225	06	00004
2	12552	10	00017
3	13467	07	00003
4	15442	06	00028
5	20554	02	00022
6	21226	06	00012
7	21654	08	00001
8	21862	11	00014
9	31577	09	00021
10	31840	01	00008
11	37192	02	00025
12	37910	04	00018
13	41623	12	00020
14	43860	07	00027
15	45878	03	00006

Pay Statement Table:

Select *
From PAYSTATEMENT

150 %

Results Messages

	EmployeeStatement#	Employee#	CompanyName	DateIssued	EmployeeSSN	BankName	AccountNumber	YTDSalary	FederalTax	StateTax	Deduction
1	00001	01	Gates	2018-01-11	852847801	Wells Fargo	652879115	10203.00	100.01	1.01	1.02
2	00002	02	Enterprise	2018-01-25	242963308	Chase	379164852	19880.00	80.05	15.58	19.75
3	00003	03	Sigma	2018-02-08	875557841	Bank of America	441977628	35010.00	58.66	21.59	6.75
4	00004	04	Gerich	2018-02-22	620414000	Citigroup	378006894	12908.00	65.77	14.22	7.25
5	00005	05	Lux	2018-03-08	621439628	Chase	121889467	22660.00	66.05	7.88	4.59
6	00006	06	Centrio	2018-03-22	220407336	Goldman Sachs	118748692	17808.00	23.44	6.23	11.88
7	00007	07	Airco	2018-04-05	413335882	Wells Fargo	307849556	9770.00	38.76	21.03	19.88
8	00008	08	Dulent	2018-04-19	299015820	Midfirst	782556781	8908.00	25.50	28.01	11.1
9	00009	09	Entrert	2018-05-03	607682253	Bank of America	674819341	11111.00	26.33	19.99	0
10	00010	10	Enclosive	2018-05-17	457357691	Midfirst	134698551	12322.00	48.70	20.55	19.88
11	00011	11	Yowell	2018-05-31	96605487	Chase	334875281	32000.00	49.23	9.74	6.5
12	00012	12	Hammurabi	2018-06-14	223075621	Chase	154979913	19999.00	42.35	5.87	0
13	00013	13	Evron	2018-06-28	110056673	Citigroup	134666973	11550.00	28.89	11.22	13.47
14	00014	14	Charlot	2018-07-12	55380752	Midfirst	557469131	17880.00	35.66	12.54	18.99
15	00015	15	Enpared	2018-07-26	771687511	Wells Fargo	306578894	9778.00	74.50	7.88	2.85
16	00016	16	Google	2018-08-09	246410226	Citigroup	997365811	10001.00	68.59	23.50	4.65
17	00017	01	Google	2018-08-23	220143279	Midfirst	975421569	11220.00	33.45	47.52	9.77
18	00018	02	IBM	2018-09-06	454178957	Chase	234619761	10011.00	27.64	12.88	11.85
19	00019	03	Sigmund	2018-09-20	608203130	Wells Fargo	154897644	18990.00	13.25	7.65	15.42
20	00020	04	Walmart	2018-10-04	143057967	Goldman Sachs	174459134	16508.00	17.98	8.67	13.87
21	00021	05	Target	2018-10-18	444035865	Citigroup	546198845	14444.00	29.59	9.01	16.66

Boarder Table:

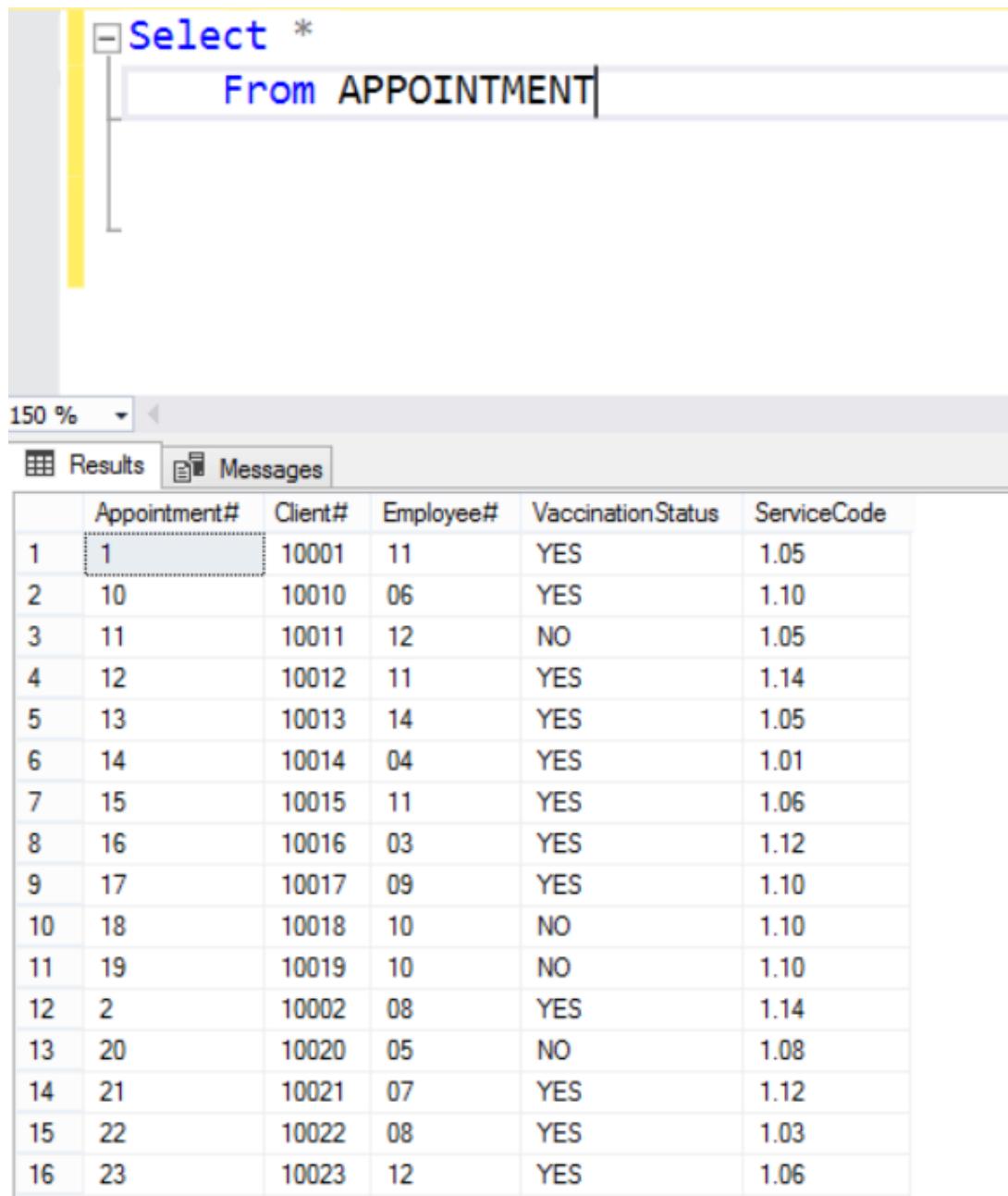
The screenshot shows a SQL query window with the following content:

```
Select *
From BOARDER
```

The results pane displays the following data:

	Employee#	Reference
1	01	John Wayne
2	03	John Travolta
3	05	Dog the Bounty Hunter

Appointment Table:



The screenshot shows a SQL query window in SSMS. The query is:

```
Select *
From APPOINTMENT
```

The results grid displays 16 rows of data from the APPOINTMENT table:

	Appointment#	Client#	Employee#	VaccinationStatus	ServiceCode
1	1	10001	11	YES	1.05
2	10	10010	06	YES	1.10
3	11	10011	12	NO	1.05
4	12	10012	11	YES	1.14
5	13	10013	14	YES	1.05
6	14	10014	04	YES	1.01
7	15	10015	11	YES	1.06
8	16	10016	03	YES	1.12
9	17	10017	09	YES	1.10
10	18	10018	10	NO	1.10
11	19	10019	10	NO	1.10
12	2	10002	08	YES	1.14
13	20	10020	05	NO	1.08
14	21	10021	07	YES	1.12
15	22	10022	08	YES	1.03
16	23	10023	12	YES	1.06

Boarded Animals Table:

The screenshot shows a SQL query window with the following content:

```
Select *
From BOARDEDANIMALS
```

The results grid displays the following data:

	Client#	ServiceCode	StayLength	CheckinDay	CheckinTime	ChecoutDay	ChecoutTime
1	10010	2.00	5	08/05/2018	10:00 AM	08/10/2018	02:50 PM
2	10026	2.00	3	08/12/2018	08:45 AM	08/15/2018	01:30 PM
3	10023	2.00	6	11/25/2018	10:00 AM	12/01/2018	5:00 PM
4	10015	2.00	6	08/23/2018	10:45 AM	08/29/2018	06:45 PM
5	10001	2.00	5	08/24/2018	12:00 PM	08/29/2018	07:00 PM
6	10016	2.00	13	08/13/2013	11:00 AM	08/26/2013	3:45 PM
7	10025	2.00	1	09/10/2018	08:00 AM	09/11/2018	05:45 PM
8	10026	2.00	2	09/25/2018	08:00 AM	09/27/2018	04:00 PM
9	10019	2.00	10	10/06/2018	09:30 AM	10/16/2018	06:00 PM
10	10006	2.00	5	10/14/2018	01:00 PM	10/19/2018	05:30 PM
11	10029	2.00	7	10/16/2018	11:15 AM	11/23/2018	03:30 PM
12	10013	2.00	14	11/08/2018	11:00 AM	11/22/2018	04:45 PM
13	10011	2.00	21	11/10/2018	10:00 AM	12/01/2018	06:45 PM
14	10006	2.00	5	12/09/2018	09:05 AM	12/14/2018	06:00 PM
15	10013	2.01	3	12/13/2018	11:30 AM	12/16/2018	06:00 PM
16	10002	2.00	3	01/05/2019	12:15 PM	01/08/2019	04:15 PM
17	10030	2.00	3	01/18/2019	03:00 PM	01/21/2019	06:15 PM

Groomed Animals Table:

The screenshot shows a SQL query window with the following content:

```
Select *
From GROOMEDANIMALS
```

The results grid displays the following data:

	Client#	ServiceCode	CurrentVaccines
1	10001	1.09	Rabies
2	10002	1.05	Rabies
3	10004	1.06	Rabies
4	10006	1.06	Rabies
5	10009	1.01	Rabies
6	10010	1.10	Rabies
7	10011	1.07	Rabies
8	10013	1.10	Rabies
9	10014	1.14	Rabies
10	10015	1.03	Rabies
11	10016	1.14	Rabies
12	10017	1.14	Rabies
13	10018	1.09	Rabies
14	10019	1.10	Rabies
15	10023	1.07	Rabies

Services Provided Table:

The screenshot shows a Microsoft SQL Server Management Studio window. At the top, there is a query editor pane with the following T-SQL code:

```
Select *
From ServicesProvided
```

Below the query editor is a results pane. The top of the results pane has a toolbar with a magnifying glass icon and a dropdown menu set to "150 %". There are two tabs: "Results" (selected) and "Messages".

	ServiceCode	ServiceType	ServiceDescription	ServiceFee
1	1.01	Grooming	Nail Trim	8.00
2	1.02	Grooming	Ear Cleaning	5.00
3	1.03	Grooming	Skin Scrape	25.00
4	1.04	Grooming	Ear Mite Treatment	10.00
5	1.05	Grooming	Flea Treatment	15.00
6	1.06	Grooming	Tick Treatment	12.50
7	1.07	Grooming	Teeth Brushing	8.50
8	1.08	Grooming	Nose & Pad Conditioning	6.00
9	1.09	Grooming	Eye Cleaning & Drops	7.50
10	1.10	Grooming	Hair Cut	15.00
11	1.11	Grooming	Chalking	4.00
12	1.12	Grooming	Stenciling	7.00
13	1.13	Grooming	Feathering	5.00
14	1.14	Grooming	Vitamins	15.00
15	1.15	Grooming	Other	30.00
16	1.16	Grooming	Matted Hair	30.00
17	2.00	Boarding	Boarding (Per Day)	45.00
18	2.01	Boarding	Medication Applications	10.00

Database Screen Captures of Queries and Results:

```
/*Names of Groomers, Their Clients, and Count of the Client */

Select E.EFName, E.ELName, C.Name, count(GA.Client#) "Count of Client"
From Employee E Join Appointment A
on E.Employee# = A.Employee#
Join Client C
on C.Client# = A.Client#
Join GroomedAnimals GA |
on GA.Client# = A.Client#
group by E.EFName, E.ELName, C.Name
```

150 % < Results Messages

	EFName	ELName	Name	Count of Client
1	Amanda	Laith	Lucky	1
2	Courtney	Alcine	Chad	1
3	Courtney	Alcine	Sparky	1
4	Dale	Timmel	Terry	1
5	Ellie	Barret	Maggie	1
6	Grace	Robertson	Annabelle	1
7	Grace	Robertson	Lars	1
8	Jonathan	Selmy	Jester	1
9	Olivia	Alaway	Pebbles	1
10	Olivia	Alaway	Puppin	1
11	Olivia	Alaway	Tiddles	1
12	Paul	Elworth	Buck	1
13	Paul	Elworth	Fabio	1
14	Riley	Shorwich	Buttons	1
15	Riley	Shorwich	Cali	1
16	Ross	Anders	Clover	1

```
/*Names of Clients in the Last Year and Date they were seen */
Select C.Name, BA.CheckInDay "Date Seen"
  From Client C Join BOARDEDANIMALS BA
    on C.Client# = BA.Client#
   Where BA.CheckInDay >= '01-01-2019'
```

150 %

	Name	Date Seen
1	Tiddles	08/05/2018
2	Pebbles	08/12/2018
3	Buck	11/25/2018
4	Chad	08/23/2018
5	Sparky	08/24/2018
6	Sprit	08/13/2018
7	Puppin	09/10/2018
8	Pebbles	09/25/2018
9	Lars	10/06/2018
10	Cali	10/14/2018
11	Terry	10/16/2018
12	Maggie	11/08/2018
13	Fabio	11/10/2018
14	Cali	12/09/2018
15	Maggie	12/13/2018
16	Bubbles	01/05/2019

```
/*Services Provided most often by species */
Select C.Species, count(A.ServiceCode) "Count of Service Codes"
    From Client C Join Appointment A
        on C.Client# = A.Client#
            Group by C.Species
```

150 %

Results Messages

	Species	Count of Service Codes
1	Cat	16
2	Dog	14

```
/*Services Provided for each client */
Select C.Name, count(A.ServiceCode) "Services Provided per Client"
    From Client C Join Appointment A
        on C.Client# = A.Client#
            Group by C.Name
                Order by count(A.ServiceCode)
```

150 %

Results Messages

	Name	Services Provided per Client
1	Annabelle	1
2	Bubbles	1
3	Buck	1
4	Buttons	1
5	Cali	1
6	Chad	1
7	Chester	1
8	Chewie	1
9	Clover	1
10	Fabio	1
11	Jazz	1
12	Jester	1
13	Lars	1
14	Lucky	1

```
[-] /*List Each Client By DOB and Weight  
[-] DoB was not recorded in our tables */  
[-] Select C.Name, C.Weight  
[-]   from CLIENT C
```

150 %

Results Messages

	Name	Weight
1	Sparky	40
2	Bubbles	22
3	Jazz	14
4	Lucky	10
5	Toby	12
6	Cali	15
7	Rotor	68
8	Tyson	15
9	Scrabs	26
10	Tiddles	38
11	Fabio	12
12	Tazzie	18
13	Maggie	16
14	Buttons	10
15	Chad	11
16	Sprit	15

```
/*List all clients by Breed */  
Select C.Name, C.Breed  
    From Client C  
    Order by C.Breed
```

150 %

	Name	Breed
1	Terry	American Bobtail
2	Bubbles	Basset Hound
3	Tazzie	Beagle
4	Cali	Bengal
5	Chester	Birman
6	Vienna	Boxer
7	Jazz	Burmese
8	Clover	Burmese
9	Buttons	Chartreux
10	Toby	Chinese Crested Dog
11	Tiddles	Chocolate Labrador
12	Puppin	English Mastiff
13	Rouge	Golden Retriever
14	Sparky	Golden Retriever
15	Rotor	Great Dane

```

/*For each animal boarded, list the client, the owner, the employee who was
working, and the number of days stay */
Select BA.Client#, C.Name, C.Owner#, O.Name,
       datediff(day, BA.CheckInDay, BA.CheckOutDay) "Number of Days Stayed"
  From BoardedAnimals BA Join Client C
    on BA.Client# = C.Client#
   Join Owner O
     on O.Owner# = C.Owner#

```

150 %

Results Messages

	Client#	Name	Owner#	Name	Number of Days Stayed
1	10010	Tiddles	00017	William Butler	5
2	10026	Pebbles	00020	Noah Wilson	3
3	10023	Buck	00027	Neal Patrick	6
4	10015	Chad	00030	Joel Jones	6
5	10001	Sparky	00002	Garcia Lopez	5
6	10016	Sprit	00006	Zach Martin	13
7	10025	Puppin	00024	Valentine Flynn	1
8	10026	Pebbles	00020	Noah Wilson	2
9	10019	Lars	00025	Tara Nelson	10
10	10006	Cali	00029	Alexandria Stevens	5
11	10029	Terry	00015	Ben Stiller	38
12	10013	Maggie	00023	Stephen Lovell	14
13	10011	Fabio	00026	Paula Steen	21
14	10006	Cali	00029	Alexandria Stevens	5
15	10013	Maggie	00023	Stephen Lovell	3
16	10002	Bubbles	00019	Morgan Hayes	3
17	10000		00010	D. J. Quill	0

```
/*Which clients have been boarded in the last month? */
Select BA.Client#, C.Name, BA.CheckInDay
  From BoardedAnimals BA Join Client C
    on BA.Client# = C.Client#
   Where BA.CheckInDay >= '10/08/2019'
```

150 %

Results Messages

	Client#	Name	CheckInDay
1	10023	Buck	11/25/2018
2	10006	Cali	10/14/2018
3	10029	Terry	10/16/2018
4	10013	Maggie	11/08/2018
5	10011	Fabio	11/10/2018
6	10006	Cali	12/09/2018
7	10013	Maggie	12/13/2018

```
/*For all clients with a boarding appointment in the next week, which do not have current vaccinations*/
Select BA.Client#, C.Name, BA.CheckInDay, G.CurrentVaccines
  From BoardedAnimals BA Join Client C
    on BA.Client# = C.Client#
   Join GROOMEDANIMALS G
     on G.Client# = C.Client#
    Where BA.CheckInDay >= '11/01/2019'
```

6 ▾

Results Messages

Client#	Name	CheckInDay	CurrentVaccines
10023	Buck	11/25/2018	Rabies
10013	Maggie	11/08/2018	Rabies
10011	Fabio	11/10/2018	Rabies
10006	Cali	12/09/2018	Rabies
10013	Maggie	12/13/2018	Rabies

```
/*Client that has been boarded the most often. */
Select BA.Client#, count(BA.Client#) "Number of Times Boarded"
    From BoardedAnimals BA
    Group by BA.Client#
    Order by count(BA.Client#) desc;
```

150 %

Results Messages

	Client#	Number of Times Boarded
1	10011	2
2	10013	2
3	10005	2
4	10006	2
5	10025	2
6	10026	2
7	10029	1
8	10030	1
9	10010	1
10	10001	1
11	10002	1
12	10004	1
13	10014	1
14	10015	1
15	10016	1

```

/*13.Which employees earn more than the average salary for all employees?*/
Select E.Employee#, E.EFName, E.ELName, E.Salary, Avg(E.Salary) "Average Salary"
From Employee E
Group By E.Employee#, E.EFName, E.ELName, E.Salary
Order by E.Salary

```

150 %

Results Messages

	Employee#	EFName	ELName	Salary	Average Salary
1	04	Riley	Showitch	23000.00	23000.000000
2	06	Olivia	Alaway	25000.00	25000.000000
3	13	Becca	Freed	25000.00	25000.000000
4	14	Ellie	Baret	25000.00	25000.000000
5	15	Nathan	Meyers	25000.00	25000.000000
6	16	David	Andrews	25000.00	25000.000000
7	09	Ross	Anders	32000.00	32000.000000
8	01	Jonathan	Selmy	42450.00	42450.000000
9	11	Courtney	Alcine	43000.00	43000.000000
10	12	Paul	Elworth	44000.00	44000.000000
11	03	Sam	Forwell	46000.00	46000.000000
12	02	Amanda	Laith	51000.00	51000.000000
13	05	Dale	Timmel	54000.00	54000.000000
14	10	Grace	Robertson	56000.00	56000.000000
15	08	Timothy	Lowell	67000.00	67000.000000
16	07	Abigail	Chavez	88000.00	88000.000000

Basic Integrity Tests:

```
/*1. Two INSERT statements in a row - the first one adding a new Client, the second one adding a new Grooming for that Client, should succeed. Basic Test #1 Before */

select *
from client

select *
from appointment

select *
from GROOMEDANIMALS
```

Results

7	10007	00008	Rotor	2014-03-06	Male	Dog	68	Great Dane	2
8	10008	00013	Tyson	2016-07-15	Female	Cat	15	Sphynx	6

Appointment#	Client#	Employee#	VaccinationStatus	ServiceCode
1	10001	11	YES	1.05
2	10	10010	06	1.10
3	11	10011	12	NO
4	12	10012	11	1.14
5	13	10013	14	YES
6	14	10014	04	1.05
7	15	10015	11	YES
8	16	10016	03	1.06
				1.12

Client#	ServiceCode	CurrentVaccines
10001	1.09	Rabies

```
/*Basic Test #1 After Screenshot. Adding new client with all information, adding appointment for client and adding that client to list of groomed animals*/
Begin transaction;

insert into Client
values ('10031', '00019', 'Sparkles', '2012-03-12', 'Female', 'Cat', 13, 'Siamese', 3)

insert into APPOINTMENT
values (00031, '10031', '08', 'YES', 1.01)

insert into GROOMEDANIMALS
values ('10031', 1.01, 'Fleas')

select *
from client
```

Results

Client#	Owner#	Name	DOB	Gender	Species	Weight	Breed	TotalOfServices
1	10031	Sparkles	2012-03-12	Female	Cat	13	Siamese	3
2	10030	00018	Jester	2016-11-04	Female	Dog	25	Old English Sheepdog
3	10029	00015	Terry	2014-02-27	Male	Cat	14	American Bobtail

Appointment#	Client#	Employee#	VaccinationStatus	ServiceCode
31	10031	08	YES	1.01
30	10030	01	YES	1.13

Client#	ServiceCode	CurrentVaccines
10031	1.01	Rabies

```

/*Basic Test #2 Before Screenshot showing tables without adding new record*/

Select *
    from client
        order by Client# desc;

Select *
    from APPOINTMENT
        order by client# desc;

select *
    from GROOMEDANIMALS
        order by client# desc;

```

150 %

	Client#	Owner#	Name	DOB	Gender	Species	Weight	Breed	Total#ofServices
1	10030	00018	Jester	2018-11-04	Female	Dog	25	Old English Sheepdog	4
2	10029	00015	Terry	2014-02-27	Male	Cat	14	American Bobtail	1
3	10028	00021	Tory	2015-04-12	Female	Cat	16	Turkish Van	9

	Appointment#	Client#	Employee#	VaccinationStatus	ServiceCode
1	30	10030	01	YES	1.13
2	29	10029	05	YES	1.14

	Client#	ServiceCode	CurrentVaccines
1	10030	1.13	Rabies
2	10029	1.03	Rabies

```

/*Basic Test #2 After Screenshot. Showing that when adding in reverse order the database fails
due to a referential integrity violation*/
Begin transaction;

insert into GROOMEDANIMALS
    values ('10031', 1.01, 'Fleas')

insert into APPOINTMENT
    values (00031, '10031', '08', 'YES', 1.01)

insert into Client
    values ('10031', '00019', 'Sparkles', '2012-03-12', 'Female', 'Cat', 13, 'Siamese', 3)

select *
    from client
        order by client# desc;

```

150 %

	Client#	ServiceCode	CurrentVaccines
1	10030	1.13	Rabies
2	10029	1.03	Rabies

Msg 547, Level 16, State 0, Line 5
The INSERT statement conflicted with the FOREIGN KEY constraint "FK_GROOMEDAN_Clien_6EC0713C". The conflict occurred.
The statement has been terminated.

Msg 547, Level 16, State 0, Line 8
The INSERT statement conflicted with the FOREIGN KEY constraint "FK_APPOINTME_Clien_671F4F74". The conflict occurred.
The statement has been terminated.

SQLQuery1.sql - ac...UAD\ranelso7 (152)* ✘ X

```

1 /* 3. Adding a new Client without gender should fail. */
2
3 select *
4 from client;
5
6
7

```

100 %

Results Messages

	Client#	Owner#	Name	DOB	Gender	Species	Weight	Breed	Total#ofServices
1	10001	00002	Sparky	2014-03-12	Male	Dog	40	Golden Retriever	1
2	10002	00019	Bubbles	2010-01-02	Male	Dog	22	Basset Hound	3
3	10003	00005	Jazz	2015-11-23	Female	Cat	14	Burmese	2
4	10004	00011	Lucky	2012-11-14	Male	Cat	10	Shorthair	4
5	10005	00001	Toby	2008-06-29	Female	Dog	12	Chinese Crested Dog	1
6	10006	00029	Cali	2017-07-19	Female	Cat	15	Bengal	2
7	10007	00008	Rotor	2014-03-06	Male	Dog	68	Great Dane	2
8	10008	00013	Tyson	2016-07-15	Female	Cat	15	Sphynx	6
9	10009	00007	Scrabs	2015-01-26	Male	Dog	26	Welsh Corgi	5
10	10010	00017	Tiddles	2015-06-11	Female	Dog	38	Chocolate Labrador	3
11	10011	00026	Fabio	2011-09-09	Male	Cat	12	Persian	4
12	10012	00004	Tazzie	2016-02-28	Male	Dog	18	Beagle	1
13	10013	00023	Maggie	2018-02-14	Male	Cat	16	Siamese	2
14	10014	00028	Buttons	2017-05-19	Male	Cat	10	Chartreux	2
15	10015	00030	Chad	2015-07-05	Female	Cat	11	Turkish Van	3
16	10016	00006	Sprit	2017-08-13	Male	Cat	15	Manx	8
17	10017	00014	Clover	2009-12-25	Female	Cat	14	Burmese	6

Query executed successfully. | acadsq17.asurite.ad.asu.edu | ASUAD\ranelso7 (152) | FA19_CIS365_82485_Team3 | 00:00:00 | 30 rows

SQLQuery1.sql - ac...UAD\ranelso7 (152)* ✘ X

```

1 /* 3. Adding a new Client without gender should fail. */
2
3 begin transaction;
4
5 INSERT INTO CLIENT (Client#, Owner#, Name, DOB, Species, Weight, Breed, Total#ofServices)
6     VALUES ('10031', '00009', 'Tabs', '2017-05-16', 'Cat', '13', 'Sphynx', 2);
7 --the above statement omits gender from the insert
8 rollback;
9
10 commit;

```

100 %

Messages

Msg 515, Level 16, State 2, Line 5
Cannot insert the value NULL into column 'Gender', table 'FA19_CIS365_82485_Team3.dbo.CLIENT'; column does not allow nulls. INSERT failed.
The statement has been terminated.

Query completed with errors. | acadsq17.asurite.ad.asu.edu | ASUAD\ranelso7 (152) | FA19_CIS365_82485_Team3 | 00:00:00 | 0 rows

```

/* Basic Test #4 Before Screenshot Showing that when looking at individual appointment
you can't see the description without Services Provided Table as well */

Select *
from APPOINTMENT

Select *
from ServicesProvided

```

150 % ▾

	Results	Messages																																																																										
1	<table border="1"> <thead> <tr> <th>1</th> <th>10001</th> <th>11</th> <th>YES</th> <th>1.05</th> </tr> </thead> <tbody> <tr><td>2</td><td>10</td><td>10010</td><td>06</td><td>YES</td></tr> <tr><td>3</td><td>11</td><td>10011</td><td>12</td><td>NO</td></tr> <tr><td>4</td><td>12</td><td>10012</td><td>11</td><td>YES</td></tr> <tr><td>5</td><td>13</td><td>10013</td><td>14</td><td>YES</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>ServiceCode</th> <th>ServiceType</th> <th>ServiceDescription</th> <th>ServiceFee</th> </tr> </thead> <tbody> <tr><td>1</td><td>1.01</td><td>Grooming</td><td>Nail Trim</td><td>8.00</td></tr> <tr><td>2</td><td>1.02</td><td>Grooming</td><td>Ear Cleaning</td><td>5.00</td></tr> <tr><td>3</td><td>1.03</td><td>Grooming</td><td>Skin Scrape</td><td>25.00</td></tr> <tr><td>4</td><td>1.04</td><td>Grooming</td><td>Ear Mite Treatment</td><td>10.00</td></tr> <tr><td>5</td><td>1.05</td><td>Grooming</td><td>Flea Treatment</td><td>15.00</td></tr> <tr><td>6</td><td>1.06</td><td>Grooming</td><td>Tick Treatment</td><td>12.50</td></tr> <tr><td>7</td><td>1.07</td><td>Grooming</td><td>Teeth Brushing</td><td>8.50</td></tr> <tr><td>8</td><td>1.08</td><td>Grooming</td><td>Nose & Pad Con...</td><td>6.00</td></tr> <tr><td>9</td><td>1.09</td><td>Grooming</td><td>Eye Cleaning & ...</td><td>7.50</td></tr> </tbody> </table>	1	10001	11	YES	1.05	2	10	10010	06	YES	3	11	10011	12	NO	4	12	10012	11	YES	5	13	10013	14	YES	ServiceCode	ServiceType	ServiceDescription	ServiceFee	1	1.01	Grooming	Nail Trim	8.00	2	1.02	Grooming	Ear Cleaning	5.00	3	1.03	Grooming	Skin Scrape	25.00	4	1.04	Grooming	Ear Mite Treatment	10.00	5	1.05	Grooming	Flea Treatment	15.00	6	1.06	Grooming	Tick Treatment	12.50	7	1.07	Grooming	Teeth Brushing	8.50	8	1.08	Grooming	Nose & Pad Con...	6.00	9	1.09	Grooming	Eye Cleaning & ...	7.50	
1	10001	11	YES	1.05																																																																								
2	10	10010	06	YES																																																																								
3	11	10011	12	NO																																																																								
4	12	10012	11	YES																																																																								
5	13	10013	14	YES																																																																								
ServiceCode	ServiceType	ServiceDescription	ServiceFee																																																																									
1	1.01	Grooming	Nail Trim	8.00																																																																								
2	1.02	Grooming	Ear Cleaning	5.00																																																																								
3	1.03	Grooming	Skin Scrape	25.00																																																																								
4	1.04	Grooming	Ear Mite Treatment	10.00																																																																								
5	1.05	Grooming	Flea Treatment	15.00																																																																								
6	1.06	Grooming	Tick Treatment	12.50																																																																								
7	1.07	Grooming	Teeth Brushing	8.50																																																																								
8	1.08	Grooming	Nose & Pad Con...	6.00																																																																								
9	1.09	Grooming	Eye Cleaning & ...	7.50																																																																								


```

/*4. For a given visit, it should be possible to see all of the descriptions
and amounts that pertain to it. Basic Test #4 After screenshot*/

```

```

create view AppointmentDescription AS /*Created View*/

select A.Appointment#, A.Client#, A.ServiceCode, SP.ServiceType, SP.ServiceDescription, SP.ServiceFee
  From Appointment A JOIN ServicesProvided SP
    on A.ServiceCode = SP.ServiceCode
  /*Joined Services Provided and Appointment to show all necessary
  data*/
select *
  from AppointmentDescription

```

150 % ▾

	Results	Messages																																																																																																							
1	<table border="1"> <thead> <tr> <th>Appointment#</th> <th>Client#</th> <th>ServiceCode</th> <th>ServiceType</th> <th>ServiceDescription</th> <th>ServiceFee</th> </tr> </thead> <tbody> <tr><td>1</td><td>10001</td><td>1.05</td><td>Grooming</td><td>Flea Treatment</td><td>15.00</td></tr> <tr><td>2</td><td>10</td><td>10010</td><td>1.10</td><td>Grooming</td><td>Hair Cut</td><td>15.00</td></tr> <tr><td>3</td><td>11</td><td>10011</td><td>1.05</td><td>Grooming</td><td>Flea Treatment</td><td>15.00</td></tr> <tr><td>4</td><td>12</td><td>10012</td><td>1.14</td><td>Grooming</td><td>Vitamins</td><td>15.00</td></tr> <tr><td>5</td><td>13</td><td>10013</td><td>1.05</td><td>Grooming</td><td>Flea Treatment</td><td>15.00</td></tr> <tr><td>6</td><td>14</td><td>10014</td><td>1.01</td><td>Grooming</td><td>Nail Trim</td><td>8.00</td></tr> <tr><td>7</td><td>15</td><td>10015</td><td>1.06</td><td>Grooming</td><td>Tick Treatment</td><td>12.50</td></tr> <tr><td>8</td><td>16</td><td>10016</td><td>1.12</td><td>Grooming</td><td>Stenciling</td><td>7.00</td></tr> <tr><td>9</td><td>17</td><td>10017</td><td>1.10</td><td>Grooming</td><td>Hair Cut</td><td>15.00</td></tr> <tr><td>10</td><td>18</td><td>10018</td><td>1.10</td><td>Grooming</td><td>Hair Cut</td><td>15.00</td></tr> <tr><td>11</td><td>19</td><td>10019</td><td>1.10</td><td>Grooming</td><td>Hair Cut</td><td>15.00</td></tr> <tr><td>12</td><td>2</td><td>10002</td><td>1.14</td><td>Grooming</td><td>Vitamins</td><td>15.00</td></tr> <tr><td>13</td><td>20</td><td>10020</td><td>1.08</td><td>Grooming</td><td>Nose & Pad Conditioning</td><td>6.00</td></tr> <tr><td>14</td><td>21</td><td>10021</td><td>1.12</td><td>Grooming</td><td>Stenciling</td><td>7.00</td></tr> </tbody> </table>	Appointment#	Client#	ServiceCode	ServiceType	ServiceDescription	ServiceFee	1	10001	1.05	Grooming	Flea Treatment	15.00	2	10	10010	1.10	Grooming	Hair Cut	15.00	3	11	10011	1.05	Grooming	Flea Treatment	15.00	4	12	10012	1.14	Grooming	Vitamins	15.00	5	13	10013	1.05	Grooming	Flea Treatment	15.00	6	14	10014	1.01	Grooming	Nail Trim	8.00	7	15	10015	1.06	Grooming	Tick Treatment	12.50	8	16	10016	1.12	Grooming	Stenciling	7.00	9	17	10017	1.10	Grooming	Hair Cut	15.00	10	18	10018	1.10	Grooming	Hair Cut	15.00	11	19	10019	1.10	Grooming	Hair Cut	15.00	12	2	10002	1.14	Grooming	Vitamins	15.00	13	20	10020	1.08	Grooming	Nose & Pad Conditioning	6.00	14	21	10021	1.12	Grooming	Stenciling	7.00	
Appointment#	Client#	ServiceCode	ServiceType	ServiceDescription	ServiceFee																																																																																																				
1	10001	1.05	Grooming	Flea Treatment	15.00																																																																																																				
2	10	10010	1.10	Grooming	Hair Cut	15.00																																																																																																			
3	11	10011	1.05	Grooming	Flea Treatment	15.00																																																																																																			
4	12	10012	1.14	Grooming	Vitamins	15.00																																																																																																			
5	13	10013	1.05	Grooming	Flea Treatment	15.00																																																																																																			
6	14	10014	1.01	Grooming	Nail Trim	8.00																																																																																																			
7	15	10015	1.06	Grooming	Tick Treatment	12.50																																																																																																			
8	16	10016	1.12	Grooming	Stenciling	7.00																																																																																																			
9	17	10017	1.10	Grooming	Hair Cut	15.00																																																																																																			
10	18	10018	1.10	Grooming	Hair Cut	15.00																																																																																																			
11	19	10019	1.10	Grooming	Hair Cut	15.00																																																																																																			
12	2	10002	1.14	Grooming	Vitamins	15.00																																																																																																			
13	20	10020	1.08	Grooming	Nose & Pad Conditioning	6.00																																																																																																			
14	21	10021	1.12	Grooming	Stenciling	7.00																																																																																																			

```
SQLQuery4.sql - ac...UAD\bgswallo (55)* + X SQLQuery3.sql - ac...UAD\bgswallo (96)*      SQLQuery2.sql - ac...UAD\bgswallo (60)*      SQLQuery1.sql - ac...AD\bgswallo (120)*
1  /* 5.Trying to delete a Client who has past visits should fail.*/
2  Begin Transaction
3
4  Delete from client
5  where client# = '10001';
6
7  rollback;
```

110 %

Messages

Msg 547, Level 16, State 0, Line 4
The DELETE statement conflicted with the REFERENCE constraint "FK_APPOINTMENT_Clien_671F4F74". The conflict occurred in database "FA19_CIS365_82485_Team3", table "dbo.APPOINTMENT", column 'Client#'.
The statement has been terminated.

```
SQLQuery3.sql - ac...UAD\bgswallo (96)* - X SQLQuery2.sql - ac...UAD\bgswallo (60)*      SQLQuery1.sql - ac...AD\bgswallo (120)*
1  /*7. Adding a new appointment without a Client# should fail. */
2
3  BEGIN TRANSACTION
4
5  INSERT INTO APPOINTMENT (Appointment#, Employee#, VaccinationStatus, ServiceCode)
6      VALUES ('31', '04', 'YES', '1.14');
7  --Statement above omits "Client#" from INSERT statement
8
9  Rollback;
10
11 Commit;
```

110 %

Messages

```
Msg 515, Level 16, State 2, Line 5
Cannot insert the value NULL into column 'Client#', table 'FA19_CIS365_82485_Team3.dbo.APPOINTMENT'; column does not allow nulls. INSERT fails.
The statement has been terminated.
```

```
/* Basic Test #8 When entering an entry into the groomer table without a school attended  
or Certificatedate the design will fail Before Screenshot*/  
  
Select *  
from GROOMER
```

150 %

	Employee#	SchoolAttended	CertificateDate	YrsOfExperience
1	07	ASU	2019-12-12	3
2	08	GCU	2018-10-12	7
3	09	NAU	2019-10-07	2
4	10	NAU	2016-02-01	5
5	11	ASU	2019-01-06	1
6	12	ASU	2019-05-06	3

```
/* Basic Test #8 When entering an entry into the groomer table without a school attended  
or Certificatedate the design will fail After Screenshot*/  
  
Begin Transaction;  
  
insert into GROOMER  
values ('13', '0')  
  
rollback:  
150 %
```

Msg 213, Level 16, State 1, Line 6
Column name or number of supplied values does not match table definition.

```

/* Basic Test #9. Two similar statements input together to bring a
new employee into the system as a part time employee but in the wrong order should fail. Before Screenshot
Showing failure*/

Select *
from Parttime

Select *
from EMPLOYEE

```

150 %

	Employee#	Reference	EmergencyContact	EmployedDate
1	13	Robert Nelson	602-444-4444	2019-12-12
2	14	John Thomas	696-696-6969	2018-02-15
3	15	Will Burr	201-432-4343	2019-05-16
4	16	Dan Wilson	708-446-5454	2018-04-15

	Employee#	EFName	ELName	Salary
1	01	Jonathan	Selmy	42450.00
2	02	Amanda	Lath	51000.00
3	03	Sam	Fonwell	46000.00
4	04	Riley	Shorwich	23000.00
5	05	Dale	Timmel	54000.00
6	06	Olivia	Alaway	25000.00
7	07	Abigail	Chavez	88000.00
8	08	Timothy	Lowell	67000.00

```

/* Basic Test #9. Two similar statements input together to bring a
new employee into the system as a part time employee but in the wrong order should fail. After Screenshot
Showing failure*/

begin transaction;
insert into parttime values ('17', 'Grace Moioli', '303-443-2948', getdate());
insert into employee values ('17', 'Clare', 'Fitzgerald', 20000);

select *
from employee;
select *
from parttime;

rollback;

```

150 %

Msg 547, Level 16, State 0, Line 6
The INSERT statement conflicted with the FOREIGN KEY constraint "FK__PARTTIME_Employ_2180FB33". The conflict occurred
The statement has been terminated.

(1 row(s) affected)

(17 row(s) affected)

(4 row(s) affected)

```

/*Basic Test #10. Creating an employee without a salary should fail. Before Screenshot
showing the values*/

select *
from EMPLOYEE

```

Results

	Employee#	EFName	ELName	Salary
1	01	Jonathan	Selmy	42450.00
2	02	Amanda	Laith	51000.00
3	03	Sam	Forwell	46000.00
4	04	Riley	Shorwich	23000.00
5	05	Dale	Timmel	54000.00
6	06	Olivia	Alaway	25000.00
7	07	Abigail	Chavez	88000.00
8	08	Timothy	Lowell	67000.00
9	09	Ross	Anders	32000.00
10	10	Grace	Robertson	56000.00
11	11	Courtney	Alcine	43000.00
12	12	Paul	Ehworth	44000.00
13	13	Becca	Freed	25000.00
14	14	Ellie	Ramet	25000.00

```

/*Basic Test #10. Creating an employee without a salary should fail. After Screenshot
showing the failure*/
begin transaction;

insert into employee (Employee#, EFName, ELName) values ('18', 'Paaris', 'Brar');

rollback;

commit;

```

Messages

```

Msg 515, Level 16, State 2, Line 5
Cannot insert the value NULL into column 'Salary', table 'FA19_CIS365_82485_Team3.dbo.EMPLOYEE'; column does not allow null
The statement has been terminated.

```

Advanced Queries:

```
/*Selecting the dog and/or dogs that visited Dip and Clip at least 1 time and were a golden retriever */  
  
select A.Client#, C.Name, count(A.Client#) "Number of Visits",  
O.Name, DATEDIFF(Year, C.DOB, GetDate()) "Client's Age"  
From Client C JOIN Owner O  
on C.Owner# = O.Owner#  
Join Appointment A  
on A.Client# = C.Client#  
Where C.Breed = 'Golden Retriever'  
Group by A.Client#, C.Name, O.Name, C.DOB  
Having count(A.Client#) >= 1
```

150 % 4

Results Messages

	Client#	Name	Number of Visits	Name	Client's Age
1	10001	Sparky	1	Garcia Lopez	5

```
SQLQuery4.sql - ac...UAD\bgswallo (55)* SQLQuery3.sql - ac...UAD\bgswallo (96)* X SQLQuery2.sql - ac...UAD\bgswallo (60)* SQLQuery1.sql - ac...AD\bgswallo (120)*
1 /*5. The total count of each provided grooming service.*/
2
3 Select A.ServiceCode, ServiceDescription, COUNT(A.ServiceCode) as "Total number of each requested services"
4     From APPOINTMENT A, ServicesProvided SP
5         Where A.ServiceCode = SP.ServiceCode
6 GROUP BY A.ServiceCode, ServiceDescription
7 Order by 3 desc;
```

Results

	ServiceCode	ServiceDescription	Total number of each requested services
1	1.10	Hair Cut	7
2	1.05	Flea Treatment	5
3	1.06	Tick Treatment	3
4	1.14	Vitamins	3
5	1.03	Skin Scrape	2
6	1.08	Nose & Pad Conditioning	2
7	1.09	Eye Cleaning & Drops	2
8	1.12	Stenciling	2
9	1.13	Feathering	2
10	1.01	Nail Trim	1
11	1.02	Ear Cleaning	1

```
SQLQuery4.sql - ac...UAD\bgswallo (55)* SQLQuery3.sql - ac...UAD\bgswallo (96)* SQLQuery2.sql - ac...UAD\bgswallo (60)* SQLQuery1.sql - ac...AD\bgswallo (120)*
1 /*3]. The total count of each species of animal that was boarded between June and July of 2018*/
2
3 Select Species, COUNT(Species) as "Total count of each species of animal that was boarded between January and April of 2019"
4   From CLIENT C, BOARDEDANIMALS BA
5     Where C.Client# = BA.Client#
6       and CheckinDay >= '01/01/2019'
7       and CheckoutDay <= '03/01/2019'
8 Group by Species;
9
```

110 %

Results Messages

Species	Total count of each species of animal that was boarded between January and April of 2019
Cat	2
Dog	3

```
SQLQuery3.sql - ac...UAD\bgswallo (96)* - X SQLQuery2.sql - ac...UAD\bgswallo (60)*      SQLQuery1.sql - ac...AD\bgswallo (120)*
1  /*4. Names of owners whose pets stayed for 2 or more days with us*/
2  Select Name
3    From OWNER
4    Where owner# in
5      (select owner#
6        from client
7        where client# in
8          (select client# from
9            BOARDEDANIMALS
10           Where staylength >'2'));
11
12
```

Results

	Name
1	Garcia Lopez
2	Zach Martin
3	Ben Stiller
4	William Butler
5	David Smith
6	Morgan Hayes
7	Noah Wilson
8	Stephen Lovell
9	Paula Steen
10	Neal Patrick
11	Alexandria Stevens
12	Joel Jones

```
SQLQuery3.sql - ac...UAD\bgswallo (96)* - SQLQuery2.sql - ac...UAD\bgswallo (60)* - SQLQuery1.sql - ac...AD\bgswallo (120)*
1  /*5. The PaycheckID for workers that went to ASU */
2  Select Employee#, paycheckID
3    From PAYCHECK
4      Where employee# in
5        (Select employee#
6          From EMPLOYEE
7              Where employee# in
8                (Select employee#
9                  From GROOMER
10                 Where schoolattended ='ASU'));
```

110 %

	Employee#	paycheckID
1	07	13467
2	11	21862
3	12	41623
4	07	43860
5	11	60213
6	11	66323
7	07	74554
8	07	74736
9	12	98945

SQLQuery2.sql - ac...UAD\nvankay2 (68)* X

```
/*The salary for the employee who worked on sparky the dog*/
select Salary
from employee
where employee# in
    (select Employee#
     from APPOINTMENT
      where Client# in
          (select client#
           from CLIENT
            where Name = 'Sparky'));
```

100 % < >

Results Messages

	Salary
1	43000.00

✓ Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\ nvankay2 (68) | FA19_CIS365_82485_Team3 | 00:00:00 | 1 rows

```

/* Advanced #7 Creating a view that shows part time employee, their first Name,
their Emergency contact, their reference, and the bank name that they are paid
out to, and salary */

Create view ParttimeInformation AS

Select PT.Employee#, E.EFName, PT.Reference, PT.EmergencyContact, PS.BankName,
E.Salary
    from Parttime PT JOIN Paystatement PS
        on PT.Employee# = PS.Employee#
        JOIN Employee E
            on PT.Employee# = E.Employee#
Select *
    from PartTimeInformation

```

150 %

	Employee#	EFName	Reference	EmergencyContact	BankName	Salary
1	13	Becca	Robert Nelson	602-444-4444	Citigroup	25000.00
2	14	Ellie	John Thomas	696-696-6969	Midfirst	25000.00
3	15	Nathan	Will Burr	201-432-4343	Wells Fargo	25000.00
4	16	David	Dan Wilson	708-446-5454	Citigroup	25000.00
5	13	Becca	Robert Nelson	602-444-4444	Goldman Sachs	25000.00
6	14	Ellie	John Thomas	696-696-6969	Midfirst	25000.00

```
/*Advanced Test #8 Creating A View THat Shows the Groomers that went to NAU  
and The Clients they worked on */  
  
Create View NAUGroomers AS  
    Select G.Employee#, E.EFName, E.ELName, A.Client#, C.Name  
        From Groomer G JOIN Employee E  
            on G.Employee# = E.Employee#  
            JOIN Appointment A  
                on G.Employee# = A.Employee#  
                JOIN Client C  
                    on A.Client# = C.Client#  
                    Where G.SchoolAttended = 'NAU'  
  
select *  
from NAUGroomers
```

Results

	Employee#	EFName	ELName	Client#	Name
1	09	Ross	Anders	10017	Clover
2	10	Grace	Robertson	10018	Annabelle
3	10	Grace	Robertson	10019	Lars
4	10	Grace	Robertson	10003	Jazz

```
/*Creating View that displays Employee's First Name, Last Name, Employee#, EmployeeStatement# Year to Date Salary from groomers that are numbered above 09 and have made less than 15,000 for the year*/  
  
create view EmployeePayStub AS  
  
Select E.EFName "Employee's First Name", E.ELName "Employee Last Name", E.Employee#, PS.EmployeeStatement#, PS.YTDSalary "Salary as of now"  
    from Employee E JOIN PayStatement PS  
        on E.Employee# = PS.Employee#  
        JOIN GROOMER G  
            on E.Employee# = G.Employee#  
            Where PS.YTDSalary < 15000  
                AND E.Employee# > 09  
  
select *  
from EmployeePayStub  
Order by [Employee Last Name];
```

	Employee's First Name	Employee Last Name	Employee#	EmployeeStatement#	Salary as of now
1	Grace	Robertson	10	00010	12322

Transaction Examples:

```

/*Section A Milestone 5 Transaction #1
Inserting a Client into the Client table using a transaction
Before Screenshot showing that client #10031 was not added*/

Select *
From CLIENT

```

150 %

Client#	Owner#	Name	DOB	Gender	Species	Weight	Breed	Total#ofServices	
18	10018	Anna...	2012-12-07	Female	Dog	22	Jack Russel Terrier	5	
19	10019	00025	Lars	2013-06-04	Female	Cat	18	Japanese Bobtail	3
20	10020	00016	Whisk...	2016-04-22	Male	Cat	14	Sphynx	2
21	10021	00009	Rouge	2016-03-09	Male	Dog	46	Golden Retriever	1
22	10022	00022	Chester	2010-01-13	Male	Cat	20	Birman	4
23	10023	00027	Buck	2017-01-29	Male	Cat	14	Korat	5
24	10024	00010	Chewie	2016-10-17	Female	Dog	13	Pomeranian	7
25	10025	00024	Puppin	2011-08-14	Female	Dog	19	English Mastiff	4
26	10026	00020	Pebbles	2012-07-27	Female	Dog	20	Rottweiler	6
27	10027	00012	Vienna	2016-04-20	Male	Dog	22	Boxer	5
28	10028	00021	Tory	2015-04-12	Female	Cat	16	Turkish Van	9
29	10029	00015	Terry	2014-02-27	Male	Cat	14	American Bobtail	1
30	10030	00018	Jester	2018-11-04	Female	Dog	25	Old English Sheep...	4

```

/*Section A Milestone 5 Transaction #1
Inserting a Client into the Client table using a transaction*/

Begin Transaction;

Insert into Client
values (10031, '00026', 'Bow Wow', '2016-08-12', 'Male', 'Dog', 17, 'Chihuahua Mix', 3)

commit;

Select *
From CLIENT

```

150 %

Client#	Owner#	Name	DOB	Gender	Species	Weight	Breed	Total#ofServices	
19	10019	00025	Lars	2013-06-04	Female	Cat	18	Japanese Bobtail	3
20	10020	00016	Whiskey	2016-04-22	Male	Cat	14	Sphynx	2
21	10021	00009	Rouge	2016-03-09	Male	Dog	46	Golden Retriever	1
22	10022	00022	Chester	2010-01-13	Male	Cat	20	Birman	4
23	10023	00027	Buck	2017-01-29	Male	Cat	14	Korat	5
24	10024	00010	Chewie	2016-10-17	Female	Dog	13	Pomeranian	7
25	10025	00024	Puppin	2011-08-14	Female	Dog	19	English Mastiff	4
26	10026	00020	Pebbles	2012-07-27	Female	Dog	20	Rottweiler	6
27	10027	00012	Vienna	2016-04-20	Male	Dog	22	Boxer	5
28	10028	00021	Tory	2015-04-12	Female	Cat	16	Turkish Van	9
29	10029	00015	Terry	2014-02-27	Male	Cat	14	American Bobtail	1
30	10030	00018	Jester	2018-11-04	Female	Dog	25	Old English Sheep...	4
31	10031	00026	Bow Wow	2016-08-12	Male	Dog	17	Chihuahua Mix	3

```

/*Section A Milestone 5 Transaction #2
When adding a new groomer employee, also add the record to the 'child'
tables to reflect all of the employee's information. Before Screenshot */

Select *
From EMPLOYEE

Select *
From GROOMER

```

150 %

		Results	Messages	
10	10	Grace	Robertson	56000.00
11	11	Courtney	Alcine	43000.00
12	12	Paul	Ehworth	44000.00
13	13	Becca	Freed	25000.00
14	14	Ellie	Barret	25000.00
15	15	Nathan	Meyers	25000.00
16	16	David	Andrews	25000.00

	Employee#	SchoolAttended	CertificateDate	YrsOfExperience
1	07	ASU	2019-12-12	3
2	08	GCU	2018-10-12	7
3	09	NAU	2019-10-07	2
4	10	NAU	2016-02-01	5
5	11	ASU	2019-01-06	1
6	12	ASU	2019-05-06	3

```

/*Section A Milestone 5 Transaction #2
When adding a new groomer employee, also add the record to the 'child'
tables to reflect all of the employee's information. After Screenshot */

Begin Transaction;

Insert into Employee
values (17, 'Tom', 'Brady', 65000)

Insert into GROOMER
values (17, 'Michigan', '2018-08-07', 1)

```

150 %

		Results	Messages	
12	12	Paul	Ehworth	44000.00
13	13	Becca	Freed	25000.00
14	14	Ellie	Barret	25000.00
15	15	Nathan	Meyers	25000.00
16	16	David	Andrews	25000.00
17	17	Tom	Brady	65000.00

	Employee#	SchoolAttended	CertificateDate	YrsOfExperience
1	07	ASU	2019-12-12	3
2	08	GCU	2018-10-12	7
3	09	NAU	2019-10-07	2
4	10	NAU	2016-02-01	5
5	11	ASU	2019-01-06	1
6	12	ASU	2019-05-06	3
7	17	Michigan	2018-08-07	1

Trigger:

The screenshot shows the SQL Server Management Studio interface with two query panes. The top pane contains the SQL code for creating a trigger:

```
1 /* Section 2 Milestone 5 Trigger. Creates trigger to increment
2   "sickdaysused" in EMPLOYEE table whenever an employee is unable
3   to come to work.*/
4
5 CREATE TRIGGER SickDayIncrement
6   ON EmployeeSickDays
7   After INSERT
8 AS
9   UPDATE EMPLOYEE
10  Set SickDaysUsed = (SickDaysUsed + 1)
11  Where Employee# =
12    (Select Employee#
13      From inserted);
```

The bottom pane shows the message "Command(s) completed successfully."

The screenshot shows the SQL Server Management Studio interface with two query panes. The top pane contains the SQL code for testing the trigger:

```
1 /* Section 2 Milestone 5 Trigger. Testing trigger to increment
2   "SickDaysUsed" in EMPLOYEE table for when employee #2 is unable
3   to come to work.*/
4
5 BEGIN TRANSACTION
6
7 Select *
8   From EMPLOYEE;
9
10 INSERT INTO EmployeeSickDays
11   VALUES( '02', '1');
12
13 Select *
14   From EMPLOYEE;
15
16 --Rollback;
```

The bottom pane shows two result sets. The first result set is a table of employee data:

	Employee#	EFName	ELName	Salary	SickDaysUsed
1	01	Jonathan	Selmy	42450.00	2
2	02	Amanda	Laith	51000.00	0
3	03	Sam	Forwell	46000.00	2
4	04	Riley	Shorwich	23000.00	2
5	05	Dale	Timmel	54000.00	0
6	06	Olivia	Alaway	25000.00	0
7	07	Abigail	Chavez	88000.00	0
8	08	Timothy	Lowell	67000.00	0

The second result set is another table of employee data, showing the change after the trigger was tested:

	Employee#	EFName	ELName	Salary	SickDaysUsed
1	01	Jonathan	Selmy	42450.00	2
2	02	Amanda	Laith	51000.00	1
3	03	Sam	Forwell	46000.00	2
4	04	Riley	Shorwich	23000.00	2
5	05	Dale	Timmel	54000.00	0

The status bar at the bottom indicates "Query executed successful...".

Procedure

```
proceduresareabitc...AD\ranelso7 (204)  ▶ X
1  /* 2.10) PROCEDURE: When a new boarding appointment is inserted, add a record to the BoardingLog and record clientID,
2   startdate, owner's name, and owners phone. Hint: You will need to create a BoardingLog to keep track of this data.
3   The procedure will fire when a new boarding appointment is scheduled. */
4
5  --this is the create table statement
6  Create table BoardingLog (
7      BoardingID int identity (1000, 5),
8      Client# char(5),
9      StartDate date,
10     OwnerName varchar(20),
11     OwnerPhone char(12))
12
13  --below is an example record to test our table
14  Insert into BoardingLog (Client#, StartDate, OwnerName, OwnerPhone)
15      values ('10010', '2018-08-05', 'Garcia Lopez', '602-623-8754')
16
17
18  --this is the code we used to create the procedure, by referencing the client# to gather the owner information
19  Create Procedure Addboardinglog
20      @Client#      integer,
21      @StartDate     date
22
23      AS
24          Insert into BoardingLog (Client#, StartDate, OwnerName, OwnerPhone)
25              Values (
26                  @client#,
27                  @startdate,
28                  (select name
29                      from owner
30                      where owner# in (
31                          select owner#
32                              from client
33                              where client# = @client#)),
34                  (select phone
35                      from owner
36                      where owner# in (
37                          select owner#
38                              from client
39                              where client# = @client#)))
40
41  --this is a test transaction to verify that the owner (Jack Bauer) was listed as he should be in the
42  -- boardinglog table, upon execution of the procedure.
43
44  begin transaction;
45
46  exec addboardinglog 10005, '2019-09-08';
47  select * from boardinglog;
48
49  rollback;
50  commit;
```

100 % ▶

Results Messages

	BoardingID	Client#	StartDate	OwnerName	OwnerPhone
1	1000	10010	2018-08-05	Garcia Lopez	602-623-8754
2	1005	10005	2019-09-08	Jack Bauer	602-555-0170

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\ranelso7 (204) | FA19_CIS365_82485_Team3 | 00:00:00 | 2 rows

Query Optimization

Pre-Optimization

```
/*Index advanced Join query used to tax the database. This query is designed to show
The Employee full name, salary count of that employee in appointments, sum of the taxes they paid,
The school that they attended for Grooming, The days that the client they helped had stayed,
The name of the client they helped. The number of days since they were certified */

set statistics time on;

Select distinct (E.EFName + E.ELName), E.Salary, count(A.employee#),
sum(PS.FederalTax + PS.StateTax) "Total Taxes", G.SchoolAttended, PS.BankName, count(BA.Client#),
Datediff (day, BA.CheckInDay, BA.CheckOutDay), C.Name, C.Client#,
DateDiff (Day, G.CertificateDate, GetDate()) "Days Since Certified"
From Employee E, Appointment A, Paystatement PS, Groomer G, Client C, BoardedAnimals BA,
ServicesProvided S
Where E.Employee# = A.Employee#
Group by E.EFName, E.ELName, E.Salary, G.SchoolAttended, PS.BankName, BA.CheckInDay, BA.CheckOutDay,
G.CertificateDate, C.Name, C.Client#, PS.YTDSalary
```

Results

(No column name)	Salary	(No column name)	Total Taxes	SchoolAttended	BankName	(No column name)	(No column name)	Name	Client#	Days Since Certified
1 AbigailChavez	88000.00	36	752.40	ASU	Wells Fargo	36	1	Annabelle	10018	324
2 AbigailChavez	88000.00	36	752.40	ASU	Wells Fargo	36	1	Bubbles	10002	-16
3 AbigailChavez	88000.00	36	752.40	ASU	Wells Fargo	36	1	Buttons	10014	204
4 AbigailChavez	88000.00	36	752.40	ASU	Wells Fargo	36	1	Cali	10006	204
5 AbigailChavez	88000.00	36	752.40	ASU	Wells Fargo	36	1	Chester	10022	204
6 AbigailChavez	88000.00	36	752.40	ASU	Wells Fargo	36	1	Chewie	10024	204
7 AbigailChavez	88000.00	36	752.40	ASU	Wells Fargo	36	1	Clover	10017	-16
8 AbigailChavez	88000.00	36	752.40	ASU	Wells Fargo	36	1	Fabio	10011	-16

```
set statistics time on;

Select distinct (E.EFName + E.ELName), E.Salary, count(A.employee#),
sum(PS.FederalTax + PS.StateTax) "Total Taxes", G.SchoolAttended, PS.BankName, count(BA.Client#),
Datediff (day, BA.CheckInDay, BA.CheckOutDay), C.Name, C.Client#,
DateDiff (Day, G.CertificateDate, GetDate()) "Days Since Certified"
From Employee E, Appointment A, Paystatement PS, Groomer G, Client C, BoardedAnimals BA,
ServicesProvided S
Where E.Employee# = A.Employee#
Group by E.EFName, E.ELName, E.Salary, G.SchoolAttended, PS.BankName, BA.CheckInDay, BA.CheckOutDay,
G.CertificateDate, C.Name, C.Client#, PS.YTDSalary
```

Results

SQL Server parse and compile time:
CPU time = 30 ms, elapsed time = 30 ms.

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.

(1053000 row(s) affected)

SQL Server Execution Times:
CPU time = 28515 ms, elapsed time = 29668 ms.

Optomizing:

```
/*Secondary index used to put together the client# and client name  
as these two are often queried together */
```

```
Create Index ClientIndex on Client (Client#, Name)
```

```
/*Secondary index used to put together the Employee first name  
and employee last name */
```

```
Create Index EmployeeIndex on Employee (EFName, ELName)
```

Optimized:

```
set statistics time on;

Select distinct E.Salary, count(A.employee#),
    sum(PS.FederalTax + PS.StateTax) "Total Taxes", G.SchoolAttended, PS.BankName, count(BA.Client#),
    Datediff (day, BA.CheckinDay, BA.CheckOutDay),
    DateDiff (Day, G.CertificateDate, GetDate()) "Days Since Certified"
From Employee E , Appointment A, Paystatement PS, Groomer G, Client C, BoardedAnimals BA,
    ServicesProvided S
    Where E.Employee# = A.Employee#
Group by E.EFName, E.ELName, E.Salary, G.SchoolAttended, PS.BankName, BA.CheckInDay, BA.CheckOutDay,
    G.CertificateDate, PS.YTDSalary
```

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.

(35100 row(s) affected)

(1 row(s) affected)

SQL Server Execution Times:
CPU time = 9719 ms, elapsed time = 9981 ms.
SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 0 ms.

```
set statistics time on;

Select distinct E.Salary, count(A.employee#),
    sum(PS.FederalTax + PS.StateTax) "Total Taxes", G.SchoolAttended, PS.BankName, count(BA.Client#),
    Datediff (day, BA.CheckinDay, BA.CheckOutDay),
    DateDiff (Day, G.CertificateDate, GetDate()) "Days Since Certified"
From Employee E , Appointment A, Paystatement PS, Groomer G, Client C, BoardedAnimals BA,
    ServicesProvided S
    Where E.Employee# = A.Employee#
Group by E.EFName, E.ELName, E.Salary, G.SchoolAttended, PS.BankName, BA.CheckInDay, BA.CheckOutDay,
    G.CertificateDate, PS.YTDSalary
```

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.

Salary	(No column name)	Total Taxes	SchoolAttended	BankName	(No column name)	(No column name)	Days Since Certified	
1	23000.00	1620	33858.00	ASU	Wells Fargo	1620	1	-16
2	23000.00	1620	33858.00	ASU	Wells Fargo	1620	1	204
3	23000.00	1620	33858.00	ASU	Wells Fargo	1620	1	324
4	23000.00	1620	33858.00	ASU	Wells Fargo	1620	2	-16
5	23000.00	1620	33858.00	ASU	Wells Fargo	1620	2	204
6	23000.00	1620	33858.00	ASU	Wells Fargo	1620	2	324

Comment on the differences between the time differences in part 2 & 4. How did you expect the result times to be different? Did you gain time, or lose time – why? (your tables may not have enough data to get a reading on the execution time – therefore, what did you expect to happen and why?)

The time decreased by about 20,000 ms when going between our original query and the second with indexes in tow. We clearly reduced the amount of data that the DBMS was having to sort through. We subtracted the time it was taking thus making the DBMS more efficient. I expected that by creating the indexes it would allow easier finding for the database and so it acted as we expected.

Views and Data Visualization

```

SQLQuery3.sql - ac...AD\bgswallo (188)* - p X SQLQuery1.sql - ac...AD\bgswallo (161)* SQLQuery2.sql - ac...UAD\bgswallo (91)*
1 /* Creating a view and data visualization table to display how much each full-time
2 employee has received in pay this year vs their total annual salary.*/
3
4
5 CREATE VIEW EmployeeYTDPay AS
6     Select FT.Employee#, AccountNumber, YTDSalary, Salary
7         From PAYSTATEMENT PS JOIN EMPLOYEE E
8             On PS.Employee# = E.Employee#
9             JOIN FULLTIME FT
10                On FT.Employee#= E.Employee#;
11
12 Select *
13     From EmployeeYTDPay;

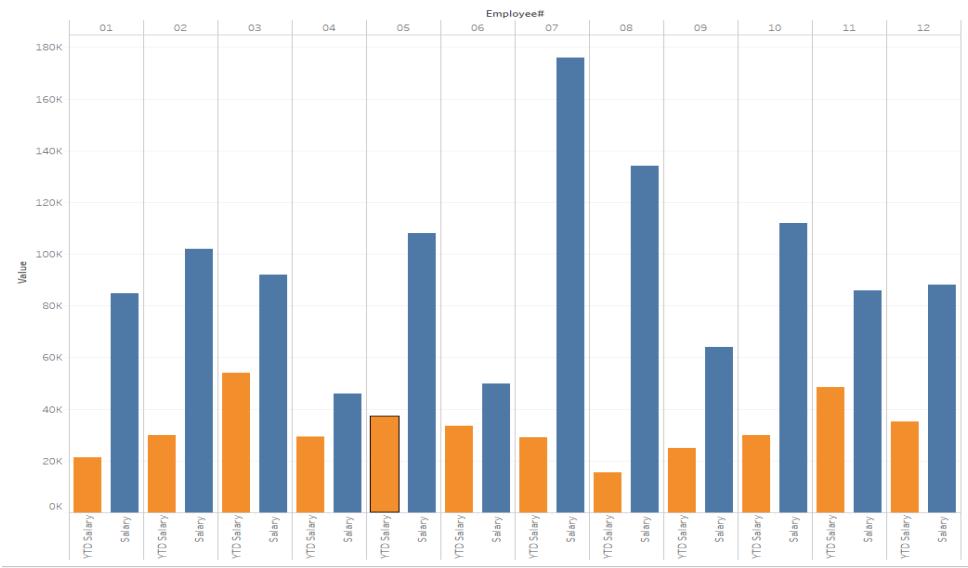
```

Results | Messages

	Employee#	AccountNumber	YTDSalary	Salary
1	01	652879115	10203.00	42450.00
2	02	379164852	19880.00	51000.00
3	03	441977628	35010.00	46000.00
4	04	378006894	12908.00	23000.00
5	05	121889467	22660.00	54000.00
6	06	118748692	17808.00	25000.00
7	07	307849556	9770.00	88000.00
8	08	782556781	8908.00	67000.00
9	09	674819341	11111.00	32000.00
10	10	134698551	12322.00	56000.00
11	11	334875281	32000.00	43000.00
12	12	154979913	19999.00	44000.00
13	01	975421569	11220.00	42450.00
14	02	234619761	10011.00	51000.00
15	03	154897644	18990.00	46000.00
16	04	174459134	16508.00	23000.00
17	05	546198845	14444.00	54000.00
18	06	723495684	15808.00	25000.00
19	07	574275411	19208.00	88000.00

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\bgswallo (188) | FA19_CIS365_82485_Team3 | 00:00:00 | 24 rows

Year to Date(YTD) Salary vs Total Annual Salary - Baden



```

/*Creating a View and then Analytical Dashboard to show all of the number of days
stayed by each client */

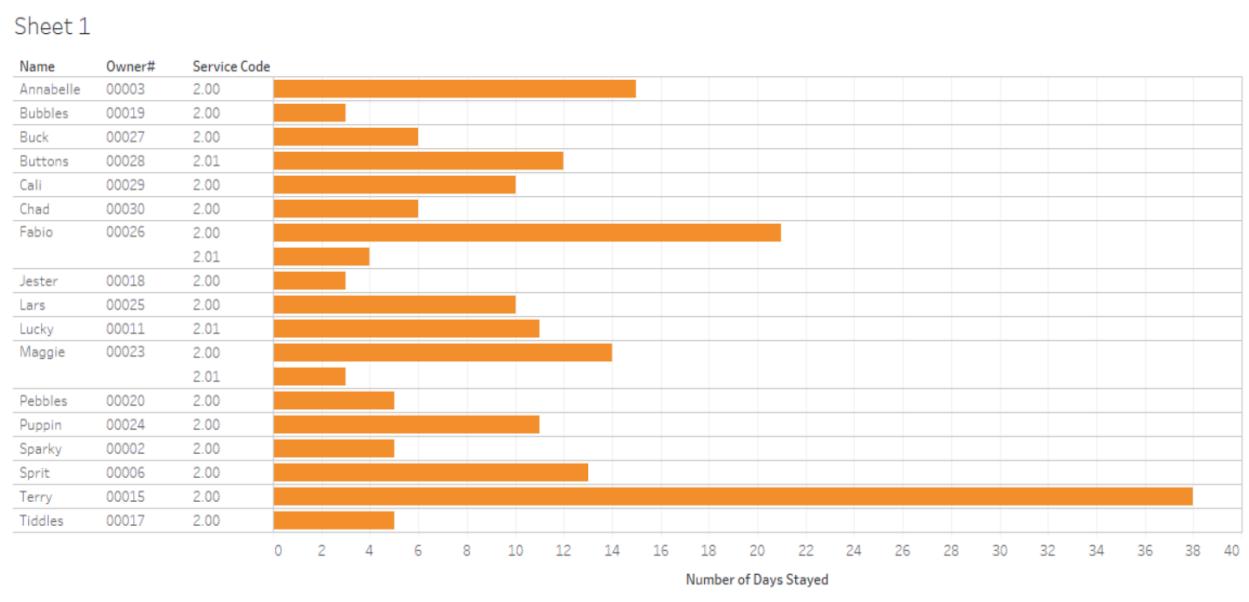
Create View TotalDaysStayed AS
Select C.Name, O.Owner#, BA.ServiceCode,
       DateDiff(day, BA.CheckInDay, BA.CheckOutDay) "Number of Days Stayed"
      From Client C Join Owner O
      On C.Owner# = O.Owner#
     Join BoardedAnimals BA
    on BA.Client# = C.Client#

Select *
  from TotalDaysStayed

```

150 %

	Name	Owner#	ServiceCode	Number of Days Stayed
1	Tiddles	00017	2.00	5
2	Pebbles	00020	2.00	3
3	Buck	00027	2.00	6
4	Chad	00030	2.00	6
5	Sparky	00002	2.00	5
6	Sprit	00006	2.00	13
7	Puppin	00024	2.00	1
8	Pebbles	00020	2.00	2
9	Lars	00025	2.00	10
10	Cali	00029	2.00	5
11	Terry	00015	2.00	38
12	Maggie	00023	2.00	14



```

SQLQuery1.sql - ac..UAD\vanelo7 (110)* # X
1 /* Create a view describing the details of a given appointment, including relevant employee, client, and service information. Include service fees. */
2
3 create view "PetShopAppointments" as
4 select appointment#, c.client#, e.employee#, EFName, ELName, Salary, Name, Gender, species, weight, sp.servicecode, servicetype, servicedescription, servicefee
5 from appointment a
6 join client c
7 on c.client# = a.client#
8 join employee e
9 on e.employee# = a.employee#
10 join servicesprovided sp
11 on sp.servicecode = a.servicecode;

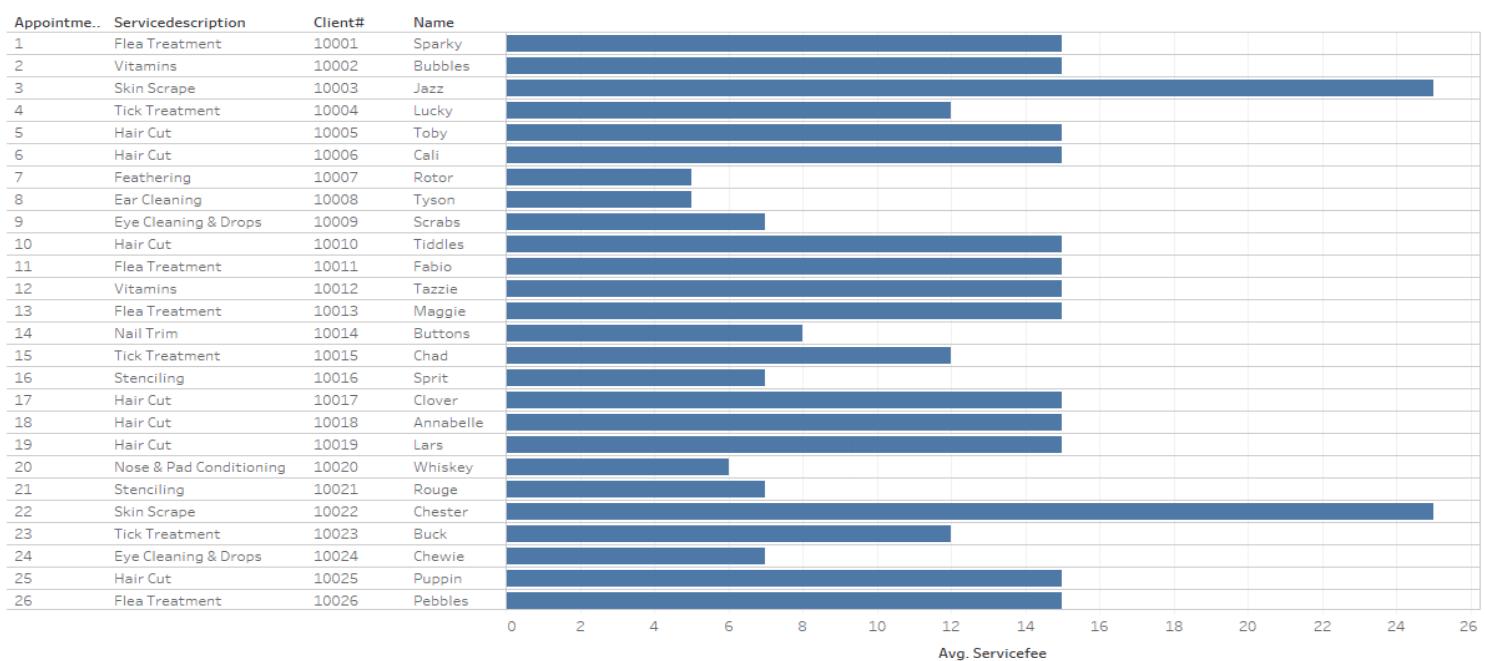
```

Results Messages

	appointment#	client#	employee#	EFName	ELName	Salary	Name	Gender	species	weight	servicecode	servicetype	servicedescription	servicefee
1	1	10001	11	Courtney	Alcine	43000.00	Sparky	Male	Dog	40	1.05	Grooming	Flea Treatment	15.00
2	10	10010	06	Olivia	Alaway	25000.00	Tiddles	Female	Dog	38	1.10	Grooming	Hair Cut	15.00
3	11	10011	12	Paul	Eworth	44000.00	Fabio	Male	Cat	12	1.05	Grooming	Flea Treatment	15.00
4	12	10012	11	Courtney	Alcine	43000.00	Tazzie	Male	Dog	18	1.14	Grooming	Vitamins	15.00
5	13	10013	14	Ellie	Barret	25000.00	Maggie	Male	Cat	16	1.05	Grooming	Flea Treatment	15.00
6	14	10014	04	Riley	Shorwich	23000.00	Buttons	Male	Cat	10	1.01	Grooming	Nail Trim	8.00
7	15	10015	11	Courtney	Alcine	43000.00	Chad	Female	Cat	11	1.06	Grooming	Tick Treatment	12.50
8	16	10016	03	Sam	Forwell	46000.00	Sprit	Male	Cat	15	1.12	Grooming	Stenciling	7.00
9	17	10017	09	Ross	Anders	32000.00	Clover	Female	Cat	14	1.10	Grooming	Hair Cut	15.00
10	18	10018	10	Grace	Robertson	56000.00	Annabelle	Female	Dog	22	1.10	Grooming	Hair Cut	15.00
11	19	10019	10	Grace	Robertson	56000.00	Lars	Female	Cat	18	1.10	Grooming	Hair Cut	15.00
12	2	10002	08	Timothy	Lowell	67000.00	Bubbles	Male	Dog	22	1.14	Grooming	Vitamins	15.00
13	20	10020	05	Dale	Timmel	54000.00	Whiskey	Male	Cat	14	1.08	Grooming	Nose & Pad Conditioning	6.00
14	21	10021	07	Abigail	Chavez	88000.00	Rouge	Male	Dog	46	1.12	Grooming	Stenciling	7.00
15	22	10002	00	Timothy	Lowell	67000.00	Chesster	Male	Cat	20	1.02	Grooming	Skin Scraps	25.00

Query executed successfully.

Cost of Grooming Services Performed - Robert



SQLQuery1.sql - ac...AD\nvankay2 (178)* ↗ ×

```

create view
mostservicesonclient
as
select Breed,Species, weight
from CLIENT;

```

100 %

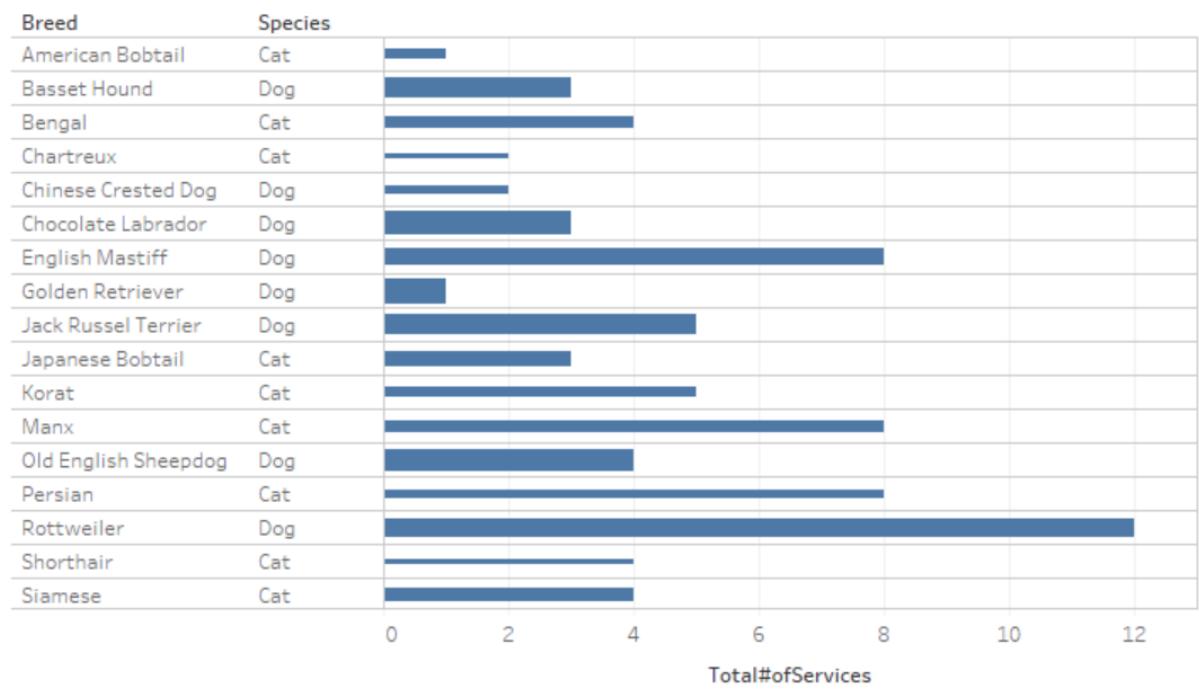
Messages

Command(s) completed successfully.

100 %

Query executed successful... | acadsql17.asurite.ad.asu.edu | ASUAD\ nvankay2 (178) | FA19_CIS365_82485_Team3 | 00:00:00 | 0 rows

Weight Vs number of services-Nayan



Major Challenges:

- Procrastination was a big issue that we faced as deadlines were fast approaching.
- We ran into several issues when we attempted to use our data in aggregate functions and so a lot of time was spent on altering data and tables.
- We found ourselves going back several times to add things that were needed by the client.
- We ran into team dynamic issues including losing a team member and having another team member that refused to cooperate or communicate.
- The entire document of asks from the client was not utilized effectively and we were tasked with providing data that we could not in the end.
- Communication was a big struggle for this team.
- Managing time effectively was also a struggle as we are a busy group and often had time management issues when it came to deadlines.

Lessons Learned:

- Take each deadline very seriously and begin to complete work ahead of time to leave time for adjusting issues.
- Issues will come with databases, but it is imperative that time is allocated to fix them.
- Make sure that database is done the right way because it will come back to haunt you.
- Get the ERD right or you will be doomed.
- Use the discussion board as much as possible with questions because a lot of them will be answered on there.

Appendix A:

```
/* Creation of Dip and Clip Database*/
```

```
create table OWNER
```

```
(Owner#      char(5),  
Name        varchar(30)    Not null,  
Address     varchar(30)    Not null,  
Phone char(10)    Not null,  
Primary key (Owner#));
```

```
Insert into OWNER values (00001, 'Jack Bauer', '54 King Ave.', 602-555-0170);
```

```
Insert into OWNER values (00002, 'Garcia Lopez', '45 Lakeview Ave.', 602-623-8754);
```

```
Insert into OWNER values (00003, 'Trisha Stevens', '348 South Tarkiln Hill St.', 480-623-4452);
```

```
Insert into OWNER values (00004, 'Alex Tills', '386 Buttonwood Ave.', 928-623-9781);
```

```
Insert into OWNER values (00005, 'Matthew Alves', '83 Park Court Rd.', 928-787-6394);
```

```
Insert into OWNER values (00006, 'Zach Martin', '383 Littleton Court', 928-891-2748);
```

```
Insert into OWNER values (00007, 'Rebecca Roberts', '12 Essex St.', 602-776-8021);
```

```
Insert into OWNER values (00008, 'Carl Grimes', '30 Glenlake Ave.', 480-995-2512);
```

```
Insert into OWNER values (00009, 'Daniel Jones', '8358 Depot Street', 480-341-5659);
```

```
Insert into OWNER values (00010, 'Enrique Vasquez', '146 Hillcrest St.', 602-415-6362);
```

```
Insert into OWNER values (00011, 'Hannah Barkley', '840 E. Roosevelt St.', 920-212-2412);
```

```
Insert into OWNER values (00012, 'Evan Jones', '22 Rock Creek St.', 623-744-6368);
```

```
Insert into OWNER values (00013, 'Julia Roberts', '8507 Sleepy Hollow St.', 602-542-8567);
```

```
Insert into OWNER values (00014, 'Ian Hamilton', '7483 Thomas Circle', 602-853-6254);
```

```
Insert into OWNER values (00015, 'Ben Stiller', '523 Peachtree Street', 480-741-9632);
```

```
Insert into OWNER values (00016, 'Lianne James', '21 North Hill Ave.', 620-852-7534);
```

```
Insert into OWNER values (00017, 'William Butler', '8793 Summit Lane', 480-846-2586);
```

```
Insert into OWNER values (00018, 'David Smith', '16 Liberty St.', 602-165-5971);
```

```
Insert into OWNER values (00019, 'Morgan Hayes', '27 N. Vermont Court', 623-734-6739);
```

```
Insert into OWNER values (00020, 'Noah Wilson', '376 Summerhouse St.', 220-563-8336);
```

```
Insert into OWNER values (00021, 'Caleb Jones', '5 West Homewood St.', 602-443-6631);
```

```
Insert into OWNER values (00022, 'Cheryl Matthews', '16 Rockaway Court', 623-623-7748);
```

```
Insert into OWNER values (00023, 'Stephen Lovell', '391 Lilac Drive', 602-881-7799);
```

```
Insert into OWNER values (00024, 'Valentine Flynn', '190 Bay Rd.', 928-552-1907);
```

```
Insert into OWNER values (00025, 'Tara Nelson', '37 NE. Rockland St.', 480-420-4480);
```

```
Insert into OWNER values (00026, 'Paula Steen', '8016 Ridgewood Rd.', 820-564-5139);
```

```

Insert into OWNER values (00027, 'Neal Patrick', '732 Hilldale Drive', 740-445-5521);
Insert into OWNER values (00028, 'Amy Stoomer', '8383 Bow Ridge St.', 602-987-3213);
Insert into OWNER values (00029, 'Alexandria Stevens', '9823 Rockcrest Road', 928-859-0738);
Insert into OWNER values (00030, 'Joel Jones', '96 10th Street', 602-823-6303);

```

create table CLIENT

```

(Client#      char(5),
Owner#       char(5),
Name         varchar(20)    Not null,
DOB          date          Not null,
Species       varchar(20)   Not null,
Weightint     Not null,
Breed         varchar(20)   Not null,
Primary key (Client#),
Foreign key (Owner#) REFERENCES OWNER(Owner#));

```

```

Insert into CLIENT values (10001, 00002, 'Sparky', '03/12/2014', 'Dog', 40, 'Golden Retriever');
Insert into CLIENT values (10002, 00019, 'Bubbles', '01/02/2010', 'Dog', 22, 'Basset Hound');
Insert into CLIENT values (10003, 00005, 'Jazz', '11/23/2015', 'Cat', 14, 'Burmese');
Insert into CLIENT values (10004, 00011, 'Lucky', '11/14/2012', 'Cat', 10, 'Shorthair');
Insert into CLIENT values (10005, 00001, 'Toby', '06/29/2008', 'Dog', 12, 'Chinese Crested Dog');
Insert into CLIENT values (10006, 00029, 'Cali', '07/19/2017', 'Cat', 15, 'Bengal');
Insert into CLIENT values (10007, 00008, 'Rotor', '03/06/2014', 'Dog', 68, 'Great Dane');
Insert into CLIENT values (10008, 00013, 'Tyson', '07/15/2016', 'Cat', 15, 'Sphynx');
Insert into CLIENT values (10009, 00007, 'Scrabs', '01/26/2015', 'Dog', 26, 'Welsh Corgi ');
Insert into CLIENT values (10010, 00017, 'Tiddles', '06/11/2015', 'Dog', 38, 'Chocolate Labrador');
Insert into CLIENT values (10011, 00026, 'Fabio', '09/09/2011', 'Cat', 12, 'Persian');
Insert into CLIENT values (10012, 00004, 'Tazzie', '02/22/2016', 'Dog', 18, 'Beagle');
Insert into CLIENT values (10013, 00023, 'Maggie', '02/14/2018', 'Cat', 16, 'Siamese');
Insert into CLIENT values (10014, 00028, 'Buttons', '05/19/2017', 'Cat', 10, 'Chartreux');
Insert into CLIENT values (10015, 00030, 'Chad', '07/05/2015', 'Cat', 11, 'Turkish Van');
Insert into CLIENT values (10016, 00006, 'Sprit', '08/13/2017', 'Cat', 15, 'Manx');
Insert into CLIENT values (10017, 00014, 'Clover', '12/25/2009', 'Cat', 14, 'Burmese');
Insert into CLIENT values (10018, 00003, 'Annabelle', '12/07/2012', 'Dog', 22, 'Jack Russel Terrier');
Insert into CLIENT values (10019, 00025, 'Lars', '06/04/2013', 'Cat', 18, 'Japanese Bobtail');
Insert into CLIENT values (10020, 00016, 'Whiskey', '04/22/2016', 'Cat', 14, 'Sphynx');

```

```

Insert into CLIENT values (10021, 00009, 'Rouge', '03/09/2016', 'Dog', 46, 'Golden Retriever');
Insert into CLIENT values (10022, 00022, 'Chester', '01/13/2010', 'Cat', 20, 'Birman');
Insert into CLIENT values (10023, 00027, 'Buck', '01/29/2017', 'Cat', 14, 'Korat');
Insert into CLIENT values (10024, 00010, 'Chewie', '10/17/2016', 'Dog', 13, 'Pomeranian');
Insert into CLIENT values (10025, 00024, 'Puppin', '08/14/2011', 'Dog', 19, 'English Mastiff');
Insert into CLIENT values (10026, 00020, 'Pebbles', '07/27/2012', 'Dog', 20, 'Rottweiler');
Insert into CLIENT values (10027, 00012, 'Vienna', '04/20/2016', 'Dog', 22, 'Boxer');
Insert into CLIENT values (10028, 00021, 'Tory', '04/12/2015', 'Cat', 16, 'Turkish Van');
Insert into CLIENT values (10029, 00015, 'Terry', '02/27/2014', 'Cat', 14, 'American Bobtail');
Insert into CLIENT values (10030, 00018, 'Jester', '11/04/2018', 'Dog', 25, 'Old English Sheepdog');

```

```

create table EMPLOYEE
(
Employee#    char(2),
EFName      varchar(15)      Not null,
ELName       varchar(15)      Not null,
Salary      decimal(8,2)      Not null,
Primary key (Employee#));

```

```

Insert into EMPLOYEE values (01, 'Jonathan', 'Selmy', 42450);
Insert into EMPLOYEE values (02, 'Amanda', 'Laith', 51000);
Insert into EMPLOYEE values (03, 'Sam', 'Forwell', 46000);
Insert into EMPLOYEE values (04, 'Riley', 'Shorwich', 23000);
Insert into EMPLOYEE values (05, 'Dale', 'Timmel', 54000);
Insert into EMPLOYEE values (06, 'Olivia', 'Alaway', 25000);
Insert into EMPLOYEE values (07, 'Abigail', 'Chavez', 88000);
Insert into EMPLOYEE values (08, 'Timothy', 'Lowell', 67000);
Insert into EMPLOYEE values (09, 'Ross', 'Anders', 32000);
Insert into EMPLOYEE values (10, 'Grace', 'Robertson', 56000);
Insert into EMPLOYEE values (11, 'Courtney', 'Alcine', 43000);
Insert into EMPLOYEE values (12, 'Paul', 'Elworth', 44000);

```

```

create table PAYSTATEMENT
(
EmployeeStatement#  char(5),
CompanyName  varchar(30),
DateIssued  date,
EmployeeSSN  char(9),

```

```

BankName varchar(30),
AccountNumber char(9),
YTDSalary decimal(4,2),
FederalTax decimal(4,2),
StateTax decimal(4,2),
Deduction decimal(4,2),
Primary key (EmployeeStatement#));

```

```
Insert into paystatement values (00001, 'Gates', '01/11/2018', 852847801, 'Wells Fargo', 652879115, 10203, 100, 1, 1);
```

```
Insert into paystatement values (00002, 'Enterprise', '01/25/2018', 242963308, 'Chase', 379164852, 19880, 80.05, 15.58, 19.75);
```

```
Insert into paystatement values (00003, 'Sigma', '02/08/2018', 875557841, 'Bank of America', 441977628, 35010, 58.66, 21.59, 6.75);
```

```
Insert into paystatement values (00004, 'Gerich', '02/22/2018', 620414000, 'Citigroup', 378006894, 12908, 65.77, 14.22, 7.25);
```

```
Insert into paystatement values (00005, 'Lux', '03/08/2018', 621439628, 'Chase', 121889467, 22660, 66.05, 7.88, 4.59);
```

```
Insert into paystatement values (00006, 'Centrio', '03/22/2018', 220407336, 'Goldman Sachs', 118748692, 17808, 23.44, 6.23, 11.88);
```

```
Insert into paystatement values (00007, 'Airco', '04/05/2018', 413335882, 'Wells Fargo', 307849556, 9770, 38.76, 21.03, 19.88);
```

```
Insert into paystatement values (00008, 'Dulent', '04/19/2018', 299015820, 'Midfirst', 782556781, 8908, 25.5, 28.01, 11.1);
```

```
Insert into paystatement values (00009, 'Entrent', '05/03/2018', 607682253, 'Bank of America', 674819341, 11111, 26.33, 19.99, 0);
```

```
Insert into paystatement values (00010, 'Enclosive', '05/17/2018', 457357691, 'Midfirst', 134698551, 12322, 48.7, 20.55, 19.88);
```

```
Insert into paystatement values (00011, 'Yowell', '05/31/2018', 96605487, 'Chase', 334875281, 32000, 49.23, 9.74, 6.5);
```

```
Insert into paystatement values (00012, 'Hammurabi', '06/14/2018', 223075621, 'Chase', 154979913, 19999, 42.35, 5.87, 0);
```

```
Insert into paystatement values (00013, 'Evron', '06/28/2018', 110056673, 'Citigroup', 134666973, 11550, 28.89, 11.22, 13.47);
```

```
Insert into paystatement values (00014, 'Charlot', '07/12/2018', 55380752, 'Midfirst', 557469131, 17880, 35.66, 12.54, 18.99);
```

```
Insert into paystatement values (00015, 'Enpared', '07/26/2018', 771687511, 'Wells Fargo', 306578894, 9778, 74.5, 7.88, 2.85);
```

```
Insert into paystatement values (00016, 'Google', '08/09/2018', 246410226, 'Citigroup', 997365811, 10001, 68.59, 23.5, 4.65);
```

```
Insert into paystatement values (00017, 'Google', '08/23/2018', 220143279, 'Midfirst', 975421569, 11220, 33.45, 47.52, 9.77);
```

```
Insert into paystatement values (00018, 'IBM', '09/06/2018', 454178957, 'Chase', 234619761, 10011, 27.64, 12.88, 11.85);
```

```
Insert into paystatement values (00019, 'Sigmund', '09/20/2018', 608203130, 'Wells Fargo', 154897644, 18990, 13.25, 7.65, 15.42);
```

```
Insert into paystatement values (00020, 'Walmart', '10/04/2018', 143057967, 'Goldman Sachs', 174459134, 16508, 17.98, 8.67, 13.87);
```

```
Insert into paystatement values (00021, 'Target', '10/18/2018', 444035865, 'Citigroup', 546198845, 14444, 29.59, 9.01, 16.66);
```

```
Insert into paystatement values (00022, 'CostCo', '11/01/2018', 234023330, 'Midfirst', 723495684, 15808, 33.7, 8.87, 23.45);  
Insert into paystatement values (00023, 'SafeWay', '11/15/2018', 136192164, 'Midfirst', 574275411, 19208, 49.81, 24.31, 4.55);  
Insert into paystatement values (00024, 'SafeWay', '11/29/2018', 199327706, 'Chase', 332447819, 6500, 99.8, 17.99, 0);  
Insert into paystatement values (00025, 'Albertsons', '12/13/2018', 262950095, 'Bank of America', 675184679, 13808, 57.22, 14.56, 5.36);  
Insert into paystatement values (00026, 'Frys Foods', '12/27/2018', 349249444, 'Citgroup', 676878512, 17544, 38.15, 11.39, 13.36);  
Insert into paystatement values (00027, 'Macys', '01/10/2019', 405901859, 'Chase', 557846199, 16433, 39.21, 19.51, 18.21);  
Insert into paystatement values (00028, 'Starbucks', '01/24/2019', 155327892, 'Chase', 987445657, 15322, 40.11, 22.58, 19.13);  
Insert into paystatement values (00029, 'Apple', '02/07/2019', 424060599, 'Goldman Sachs', 842849301, 18488, 27.54, 24.57, 8.77);  
Insert into paystatement values (00030, 'Amazon', '02/21/2019', 222942933, 'Midfirst', 787763921, 20050, 47.56, 23.65, 6.32);
```

```
create table PAYCHECK  
(PaycheckID char(5),  
Primary key (PaycheckID),  
Employee# char(2),  
EmployeeStatement# char(5),  
Foreign key (EmployeeStatement#) REFERENCES PAYSTATEMENT (EmployeeStatement#),  
Foreign key (Employee#) REFERENCES EMPLOYEE(Employee#));
```

```
Insert into paycheck values (21654, 08, 48541);  
Insert into paycheck values (96641, 04, 31759);  
Insert into paycheck values (13467, 07, 79595);  
Insert into paycheck values (11225, 06, 87757);  
Insert into paycheck values (96854, 09, 78603);  
Insert into paycheck values (45878, 03, 60296);  
Insert into paycheck values (66391, 02, 68453);  
Insert into paycheck values (31840, 01, 94793);  
Insert into paycheck values (98945, 12, 62590);  
Insert into paycheck values (66323, 11, 65290);  
Insert into paycheck values (74554, 07, 19081);  
Insert into paycheck values (21226, 06, 94649);  
Insert into paycheck values (62448, 08, 53617);
```

```
Insert into paycheck values (21862, 11, 26533);
Insert into paycheck values (66558, 03, 93389);
Insert into paycheck values (48633, 05, 64551);
Insert into paycheck values (12552, 10, 70439);
Insert into paycheck values (37910, 04, 55985);
Insert into paycheck values (56481, 06, 43031);
Insert into paycheck values (41623, 12, 12286);
Insert into paycheck values (31577, 09, 82842);
Insert into paycheck values (20554, 02, 68413);
Insert into paycheck values (74736, 07, 52893);
Insert into paycheck values (94544, 04, 33930);
Insert into paycheck values (37192, 02, 65600);
Insert into paycheck values (60213, 11, 20797);
Insert into paycheck values (43860, 07, 42431);
Insert into paycheck values (15442, 06, 88576);
```

```
create table FULLTIME
(Employee char(2),
Employee# char(2)
Foreign key (Employee#) REFERENCES EMPLOYEE(Employee#));
```

```
Insert into FULLTIME values (01);
Insert into FULLTIME values (02);
Insert into FULLTIME values (03);
Insert into FULLTIME values (04);
Insert into FULLTIME values (05);
Insert into FULLTIME values (06);
Insert into FULLTIME values (07);
Insert into FULLTIME values (08);
Insert into FULLTIME values (09);
Insert into FULLTIME values (10);
Insert into FULLTIME values (11);
Insert into FULLTIME values (12);
```

```
create table BOARDER
(Employee char(2),
Employee# char(2),
EReferences varchar(30),
Foreign key (Employee#) REFERENCES EMPLOYEE(Employee#));
```

```
Insert into BOARDER values (3, 'Micheal Leery');
```

```
Insert into BOARDER values (4, 'Sam Welsh');
```

```
Insert into BOARDER values (5, 'Matt Stone');
```

```
Insert into BOARDER values (6, 'Vanessa Rothworth');
```

```
create table GROOMER
(SchoolAttended varchar(30),
CertificateDate date,
YrsOfExperience int,
Employee# char(2)
Foreign key (Employee#) REFERENCES EMPLOYEE(Employee#));
```

```
Insert into GROOMER values (7, 'ASU', '12/12/2019', 3);
```

```
Insert into GROOMER values (8, 'GCU', '10/12/2018', 7);
```

```
Insert into GROOMER values (9, 'NAU', '10/07/2019', 2);
```

```
Insert into GROOMER values (10, 'NAU', '02/01/2016', 5);
```

```
Insert into GROOMER values (11, 'ASU', '01/06/2019', 1);
```

```
Insert into GROOMER values (12, 'ASU', '05/06/2019', 3);
```

```
create table RECORDKEEPER
(Employee# char(2),
Degree varchar(30),
GradDate date,
YrsOfExperience varchar(2),
Foreign key (Employee#) REFERENCES EMPLOYEE(Employee#));
```

```
Insert into RECORDKEEPER values (1, 'CIS', '12/12/2019', 3);
```

```
Insert into RECORDKEEPER values (2, 'FIN', '10/12/2018', 4);
```

```
create table PARTTIME
```

```

(Employee#    char(2),
EReferences   varchar(30)      Not null,
EmergencyContact  char(12) Not null,
EmployeeDate Date      Not null,
Foreign Key(Employee#) REFERENCES Employee(Employee#));

```

```

Insert into parttime values (13, 'Robert Nelson', '602-444-4444', '12/12/2019');
Insert into parttime values (14, 'John Thomas', '696-696-6969', '02/15/2018');
Insert into parttime values (15, 'Will Burr', '201-432-4343', '05/16/2019');
Insert into parttime values (16, 'Dan Wilson', '708-446-5454', '04/15/2018');

```

```

create table ServicesProvided
(ServiceCode   decimal(2,2),
ServiceType   varchar(30)      Not null,
ServiceFee    decimal(3,2)      Not null,
ServiceDescription  varchar(30)      Not null,
Primary Key(ServiceCode));

```

```

Insert into SERVICESPROVIDED values (1.01, 'Grooming', 'Nail Trim', 8.00);
Insert into SERVICESPROVIDED values (1.02, 'Grooming', 'Ear Cleaning', 5.00);
Insert into SERVICESPROVIDED values (1.03, 'Grooming', 'Skin Scrape', 25.00);
Insert into SERVICESPROVIDED values (1.04, 'Grooming', 'Ear Mite Treatment', 10.00);
Insert into SERVICESPROVIDED values (1.05, 'Grooming', 'Flea Treatment', 15.00);
Insert into SERVICESPROVIDED values (1.06, 'Grooming', 'Tick Treatment', 12.50);
Insert into SERVICESPROVIDED values (1.07, 'Grooming', 'Teeth Brushing', 8.50);
Insert into SERVICESPROVIDED values (1.08, 'Grooming', 'Nose & Pad Conditioning', 6.00);
Insert into SERVICESPROVIDED values (1.09, 'Grooming', 'Eye Cleaning & Drops', 7.50);
Insert into SERVICESPROVIDED values (1.10, 'Grooming', 'Hair Cut', 15.00);
Insert into SERVICESPROVIDED values (1.11, 'Grooming', 'Chalking', 4.00);
Insert into SERVICESPROVIDED values (1.12, 'Grooming', 'Stenciling', 7.00);
Insert into SERVICESPROVIDED values (1.13, 'Grooming', 'Feathering', 5.00);
Insert into SERVICESPROVIDED values (1.14, 'Grooming', 'Vitamins', 15.00);
Insert into SERVICESPROVIDED values (1.15, 'Grooming', 'Other', 30.00);
Insert into SERVICESPROVIDED values (1.16, 'Grooming', 'Matted Hair', 30.00);
Insert into SERVICESPROVIDED values (2.00, 'Grooming', 'Boarding (Per Day)', 45.00);
Insert into SERVICESPROVIDED values (2.01, 'Boarding', 'Medication Applications', 10.00);

```

```

create table APPOINTMENT
(
Appointment# char(5),
Client#char(5) Not null,
Employee#     char(2) Not null,
VaccinationStatus      varchar(3)      Not null,
ServiceCode    decimal(2,2)    Not null,
Primary Key(Appointment#),
Foreign Key(Client#) REFERENCES Client(Client#),
Foreign Key(Employee#) REFERENCES Employee(Employee#),
Foreign Key(ServiceCode) REFERENCES ServicesProvided(ServiceCode));

```

```

Insert into APPOINTMENT values (00001, 10001, 11, 'YES', 1.05);
Insert into APPOINTMENT values (00002, 10002, 08, 'YES', 1.14);
Insert into APPOINTMENT values (00003, 10003, 10, 'YES', 1.03);
Insert into APPOINTMENT values (00004, 10004, 02, 'YES', 1.06);
Insert into APPOINTMENT values (00005, 10005, 01, 'YES', 1.10);
Insert into APPOINTMENT values (00006, 10006, 04, 'YES', 1.10);
Insert into APPOINTMENT values (00007, 10007, 04, 'YES', 1.13);
Insert into APPOINTMENT values (00008, 10008, 07, 'NO', 1.02);
Insert into APPOINTMENT values (00009, 10009, 03, 'YES', 1.09);
Insert into APPOINTMENT values (00010, 10010, 06, 'YES', 1.10);
Insert into APPOINTMENT values (00011, 10011, 12, 'NO', 1.05);
Insert into APPOINTMENT values (00012, 10012, 11, 'YES', 1.14);
Insert into APPOINTMENT values (00013, 10013, 14, 'YES', 1.05);
Insert into APPOINTMENT values (00014, 10014, 04, 'YES', 1.01);
Insert into APPOINTMENT values (00015, 10015, 11, 'YES', 1.06);
Insert into APPOINTMENT values (00016, 10016, 03, 'YES', 1.12);
Insert into APPOINTMENT values (00017, 10017, 09, 'YES', 1.10);
Insert into APPOINTMENT values (00018, 10018, 10, 'NO', 1.10);
Insert into APPOINTMENT values (00019, 10019, 10, 'NO', 1.10);
Insert into APPOINTMENT values (00020, 10020, 05, 'NO', 1.08);
Insert into APPOINTMENT values (00021, 10021, 07, 'YES', 1.12);
Insert into APPOINTMENT values (00022, 10022, 08, 'YES', 1.03);
Insert into APPOINTMENT values (00023, 10023, 12, 'YES', 1.06);
Insert into APPOINTMENT values (00024, 10024, 02, 'YES', 1.09);

```

```

Insert into APPOINTMENT values (00025, 10025, 06, 'YES', 1.10);
Insert into APPOINTMENT values (00026, 10026, 06, 'YES', 1.05);
Insert into APPOINTMENT values (00027, 10027, 03, 'YES', 1.05);
Insert into APPOINTMENT values (00028, 10028, 11, 'YES', 1.08);
Insert into APPOINTMENT values (00029, 10029, 05, 'YES', 1.14);
Insert into APPOINTMENT values (00030, 10030, 01, 'YES', 1.13);

```

```

create table BOARDEDANIMALS
(
Client# char(5),
ServiceCode decimal(2,2) Not null,
StayLength varchar(2) Not null,
CheckinDay date Not null,
CheckinTime char(8) Not null,
CheckoutDay date Not null,
CheckoutTime char(8) Not null,
Foreign Key(Client#) REFERENCES Client(Client#),
Foreign Key(ServiceCode) REFERENCES ServicesProvided(ServiceCode))

```

```

Insert into BOARDEDANIMALS values (2.00, 2.00, 05, '08/05/2018', '10:00 AM', '08/10/2018', '02:50 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 03, '08/12/2018', '08:45 AM', '08/15/2018', '01:30 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 09, '08/14/2018', '08:25 AM', '08/20/2018', '03:25 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 06, '08/23/2018', '10:45 AM', '08/29/2018', '06:45 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 05, '08/24/2018', '12:00 PM', '08/29/2018', '07:00 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 05, '09/01/2018', '10:20 AM', '09/06/2018', '05:30 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 01, '09/10/2018', '08:00 AM', '09/11/2018', '05:45 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 02, '09/25/2018', '08:00 AM', '09/27/2018', '04:00 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 10, '10/06/2018', '09:30 AM', '10/16/2018', '06:00 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 05, '10/14/2018', '01:00 PM', '10/19/2018', '05:30 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 07, '10/16/2018', '11:15 AM', '11/23/2018', '03:30 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 14, '11/08/2018', '11:00 AM', '11/22/2018', '04:45 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 21, '11/10/2018', '10:00 AM', '12/01/2018', '06:45 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 06, '11/25/2018', '10:00 AM', '11/31/2018', '05:00 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 05, '12/09/2018', '09:05 AM', '12/14/2018', '06:00 PM');
Insert into BOARDEDANIMALS values (2.00, 2.01, 03, '12/13/2018', '11:30 AM', '12/16/2018', '06:00 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 03, '01/05/2019', '12:15 PM', '01/08/2019', '04:15 PM');
Insert into BOARDEDANIMALS values (2.00, 2.00, 03, '01/18/2019', '03:00 PM', '01/21/2019', '06:15 PM');

```

```
Insert into BOARDEDANIMALS values (2.00, 2.00, 15, '01/23/2019', '10:25 AM', '02/07/2019', '05:50 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.01, 12, '02/14/2019', '09:00 AM', '02/26/2019', '04:35 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.01, 08, '02/20/2019', '09:00 AM', '02/28/2019', '06:00 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.00, 10, '03/10/2019', '09:00 AM', '03/20/2019', '05:30 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.01, 04, '03/14/2019', '08:15 AM', '03/18/2019', '04:00 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.02, 07, '04/02/2019', '02:30 PM', '04/09/2019', '07:00 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.03, 07, '04/21/2019', '11:45 AM', '04/28/2019', '05:30 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.04, 02, '05/17/2019', '10:45 AM', '05/19/2019', '05:00 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.05, 16, '06/06/2019', '02:00 PM', '05/22/2019', '07:30 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.06, 07, '06/19/2019', '01:00 PM', '06/26/2019', '06:45 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.07, 18, '06/29/2019', '10:00 AM', '07/17/2019', '05:30 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.08, 10, '07/04/2019', '09:30 AM', '07/14/2019', '06:10 PM');  
Insert into BOARDEDANIMALS values (2.00, 2.01, 12, '07/11/2019', '08:45 AM', '07/22/2019', '05:50 PM');
```

```
create table GroomedAnimals
```

```
(Client#      char(5),  
ServiceCode    decimal(2,2)    Not null,  
CurrentVaccines    varchar(30)    Not null  
Foreign Key(Client#) REFERENCES Client(Client#),  
Foreign Key(ServiceCode) REFERENCES ServicesProvided(ServiceCode))
```

```
Insert into GROOMEDANIMALS values (10010, 1.10, 'Rabies');  
Insert into GROOMEDANIMALS values (10026, 1.16, 'Rabies');  
Insert into GROOMEDANIMALS values (10023, 1.07, 'Rabies');  
Insert into GROOMEDANIMALS values (10015, 1.03, 'Rabies');  
Insert into GROOMEDANIMALS values (10001, 1.09, 'Rabies');  
Insert into GROOMEDANIMALS values (10016, 1.14, 'Rabies');  
Insert into GROOMEDANIMALS values (10025, 1.13, 'Rabies');  
Insert into GROOMEDANIMALS values (10026, 1.10, 'Rabies');  
Insert into GROOMEDANIMALS values (10019, 1.10, 'Rabies');  
Insert into GROOMEDANIMALS values (10006, 1.06, 'Rabies');  
Insert into GROOMEDANIMALS values (10029, 1.03, 'Rabies');  
Insert into GROOMEDANIMALS values (10013, 1.10, 'Rabies');  
Insert into GROOMEDANIMALS values (10011, 1.07, 'Rabies');  
Insert into GROOMEDANIMALS values (10023, 1.09, 'Rabies');  
Insert into GROOMEDANIMALS values (10006, 1.01, 'Rabies');
```

```
Insert into GROOMEDANIMALS values (10013, 1.08, 'Rabies');
Insert into GROOMEDANIMALS values (10002, 1.05, 'Rabies');
Insert into GROOMEDANIMALS values (10030, 1.13, 'Rabies');
Insert into GROOMEDANIMALS values (10018, 1.09, 'Rabies');
Insert into GROOMEDANIMALS values (10014, 1.14, 'Rabies');
Insert into GROOMEDANIMALS values (10016, 1.03, 'Rabies');
Insert into GROOMEDANIMALS values (10025, 1.05, 'Rabies');
Insert into GROOMEDANIMALS values (10011, 1.10, 'Rabies');
Insert into GROOMEDANIMALS values (10013, 1.15, 'Rabies');
Insert into GROOMEDANIMALS values (10006, 1.14, 'Rabies');
Insert into GROOMEDANIMALS values (10027, 1.07, 'Rabies');
Insert into GROOMEDANIMALS values (10029, 1.12, 'Rabies');
Insert into GROOMEDANIMALS values (10010, 1.06, 'Rabies');
Insert into GROOMEDANIMALS values (10017, 1.14, 'Rabies');
Insert into GROOMEDANIMALS values (10009, 1.01, 'Rabies');
Insert into GROOMEDANIMALS values (10004, 1.06, 'Rabies');
```

```
*****
*****
```

```
/*Basic Test #1 Two Insert Statements in a row - The first one adding a new client,
the second one adding a new appointment for that new client, and a new record
of that client being groomed Before Screenshot*/
```

```
Select *
from client
```

```
select *
from appointment
```

```
select *
from groomedanimals
```

```
/*Basic Test #1 Two Insert Statements in a row - The first one adding a new client,
```

the second one adding a new appointment for that new client, and a new record
of that client being groomed After Screenshot*/

Begin Transaction;

Insert into Client

```
values ('10031', '00019', 'Sparkles', '2012-03-12', 'Female',  
'Cat', 13, 'Siamese', 3)
```

Insert into Appointment

```
values (00031, '10031', '08', 'YES', 1.01)
```

Insert into GroomedAnimals

```
values ('10031', 1.01, 'Fleas')
```

select *

from client

select *

from appointment

select *

from groomedanimals

/*Basic Test #2 Before screenshot showing tables without adding new record */

select *

from client

order by client# desc;

select *

from appointment

order by client# desc;

select *

```
from groomedanimals
    order by client# desc;

/*Basic Test #2 After Screenshot showing that when adding in reverse order
the database fails due to referential integrity violation */
```

```
Begin Transaction;
```

```
Insert into GroomedAnimals
    values ('10031', 1.01, 'Fleas'
```

```
Insert into Appointment
    values (00031, '10031', '08', 'YES', 1.01)
```

```
Insert into client
    values ('10031', '00019', 'Sparkles', '2012-03-12', 'Female', 'Cat',
           , 13, 'Siamese', 3)
```

```
Select *
    from Client
        order by Client# desc;
```

```
/*Basic Test #3 Adding a new Client without gender should fail
Before Screenshot */
```

```
Select *
    from client;

/*Basic Test #3 Adding a new Client without gender should fail
After Screenshot */
```

```
Begin Transaction;
```

```
Insert into Client (Client#, Owner#, Name, DOB, SPecies, Weight, Breed, Total#ofServices)
    Values ('10031', '00009', 'Tabs', '2017-05-16', 'Cat', '13', 'Sphynx', 2);
```

```
rollback;

commit;

/*Basic Test #4 Before screenshot Showing that when looking at individual appointment
you can't see the description without Services provided table as well */
```

```
Select *
from Appointment
```

```
Select *
from ServicesProvided
```

```
/*Basic Test #4 For a given visit, it should be possible to see all of the
descriptions and amounts that pertain to it. Basic test #4 after screenshot */
```

```
Create view AppointmentDescription AS /*Created View*/
```

```
Select A.Appointment#, A.Client#, A.ServiceCode, SP.ServiceType, SP.ServiceDescription,
SP.ServiceFee
From Appointment A JOIN ServicesProvided SP
on A.ServiceCode = SP.ServiceCode
/*Joined Services Provided and Appointment to show all necessary
data */
```

```
Select *
from AppointmentDescription
```

```
/*Basic Test #5 Trying to delete a client who has past visits should fail */
```

```
Begin Transaction;
```

```
delete from client
where client# = '10001'
```

```
rollback;
```

```
/*Basic Test #7 Adding a new appointment without a client# should fail */
```

```
Begin Transaction
```

```
Insert into Appointment (Appointment#, EMployee#, VaccinationStatus, ServiceCode)
```

```
    Values ('31, '04', 'YES', '1.14');
```

```
--Statement above omits Client# from Insert Statement
```

```
Rollback;
```

```
Commit;
```

```
/* Basic Test #8 When entering an entry into the groomer table without a school  
attended or certificate the design will faile before screenshot */
```

```
Select *
```

```
From Groomer
```

```
/* Basic Test #8 When entering an entry into the groomer table without a school  
attended or certificatedate the design will fail after screenshot */
```

```
Begin Transaction;
```

```
Insert into Groomer
```

```
values ('13', '0')
```

```
rollback;
```

```
/* Basic Test #9. Two similar statements input together to bring a new employee  
into the system as a part time employee but in the wrong order should fail.  
Before screenshot showing failure Before Screenshot*/
```

```
Select *  
from Parttime
```

```
Select *  
from Employee
```

```
/* Basic Test #9. Two Similar statements input together to bring a new employee  
into the system as a part time employee but in the wrong order should fail.
```

```
After screenshot showing failure */
```

```
Begin Transaction;  
Insert into parttime values ('17', 'Grace Moioli', '303-443-2948', getdate());  
Insert into employee values ('17', 'Clare', 'Fitzgerald', 20000);
```

```
select *  
from employee;  
select *  
from parttime;
```

```
rollback;
```

```
/* Basic Test #10. Creating an employee without a salary should fail. Before screenshot  
showing the values Before Screenshot*/
```

```
Select *  
from Employee
```

```
/*Basic Test #10. Creating an employee without a salary should fail. After screenshot  
showing the failure */
```

```
begin transaction;
```

```
insert into employee (employee#, EFName, ELName) values ('18', 'Paaris', 'Brar');
```

```
rollback;
```

```
commmit;
```

```
/*Selecting the dog and/or dogs that visited Dip and Clip at least 1 time and were  
a golden retriever Advanced #1*/
```

```
Select A.Client#, C.Name, count(A.Client#) "Number of Visits",  
O.Name, DateDiff(Year, C.DOB, GetDate()) "Client's Age"  
    From Client C JOIN Owner O  
        on C.Owner# = O.Owner#  
    JOIN Appointment A  
        on A.Client# = C.Client#  
    Where C.Breed = 'Golden Retriever'  
    Group by A.Client#, C.Name, O.Name, C.DOB  
    Having count(A.Client#) >= 1
```

```
/*Advanced #2 The total count of each provided grooming service */
```

```
Select A.ServiceCode, ServiceDescription, Count(A.ServiceCode) as "Total Number of each requested services"  
    From Appointment A, ServicesProvided SP  
        Where A.ServiceCode = SP.ServiceCode  
    Group by A.ServiceCode, ServiceDescription  
    Order by 3 desc;
```

```
/* Advanced #3 The total count of each species of animal that was boarded between  
June and July of 2018 */
```

```
Select Species, Count(Species) as "Total count of each species of animal that was boarded between January and  
April of 2019"  
    From Client C, BoardedAnimals BA  
        Where C.Client# = BA.Client#  
            and CheckInDay >= '01/01/2019'  
            and CheckoutDay <= '03/01/2019'  
    Group by species;
```

```
/*Advanced #4 Names of owners whose pets stayed for 2 or more days with us */
```

Select Name

From Owner

Where owner# in

(Select owner#

from client

where client# in

(select client# from

BOARDEDANIMALS

Where staylength > '2'));

```
/* Advanced #5 The paycheckID for workers that went to ASU */
```

Select Employee#, paycheckID

from PAYCHECK

Where employee# In

(Select employee#

From Employee

Where Employee# in

(Select employee#

from Groomer

Where schoolattended = 'ASU'));

```
/* Advanced #6 The Salary for the employee who worked on sparky the dog */
```

Select Salary

from employee

where employee# in

(select Employee#

from APPOINTMENT

where Client# in

(select Client#

from CLIENT

where Name = 'SParky'));

```
/* Advanced #7 Creating a view that shows part time employee, their first Name,  
their Emergency contact, their reference, and the bank name that they are paid  
out to, and salary */
```

Create view ParttimeInformation AS

```
Select PT.Employee#, E.EFName, PT.Reference, PT.EmergencyContact, PS.BankName,  
E.Salary
```

```
from Parttime PT JOIN Paystatement PS  
on PT.Employee# = PS.Employee#  
JOIN Employee E  
on PT.Employee# = E.Employee#
```

Select *

```
from PartTimeInformation
```

```
/*Advanced Test #8 Creating a View that shows the groomers that went to NAU  
and the clients they worked on */
```

Create View NAUGroomers AS

```
Select G.Employee#, E.EFName, A.Client#, C.Name  
From Groomer G JOIN Employee E  
on G.Employee# = A.Employee#  
JOIN Client C  
on A.Client# = C.Client#  
Where G.SchoolAttended = 'NAU'
```

Select *

```
from NAUGroomers
```

```
/*Creating view that displays employee's first name, last name, employee#, employeestatement#  
Year to date salary from groomers that are numbered above 09 and have made less  
than 15,000 for YTD */
```

Create view EmployeePayStub AS

```
Select E.EFName "Employee's First Name", E.ELName "Employee Last Name", E.Employee#.
```

```
PS.EmployeeStatement#, PS.YTDSalary "Salary as of Now"  
from Employee E JOIN Paystatement PS  
on E.Employee# = PS.Employee#  
JOIN Groomer G  
on E.Employee# = G.Employee#  
Where PS.YTDSalary < 15000  
AND E.Employee# > 09
```

```
Select *  
from EmployeePaystub  
Order by [Employee Last Name];
```

```
/*Advanced Test #7 Creating A View that shows part time employee, Their First Name,  
their Emegergency contact, Their Reference and the Bank Name that they are paid out to, and salary */
```

```
select *  
from Parttime with
```

```
Create View PartTimeInformation AS
```

```
select PT.Employee#, E.EFName, PT.Reference, PT.EmergencyContact, PS.BankName, E.Salary  
from Parttime PT JOIN PAYSTATEMENT PS  
on PT.Employee# = PS.Employee#  
JOIN Employee E  
on PT.Employee# = E.Employee#
```

```
select *  
from PartTimeInformation with
```

```
*****  
*****
```

```
/*Team3-M5 Master Text File*/
```

```
/*Section A Milestone 5 Transaction #1
```

```
Inserting a Client into the Client table using a transaction*/
```

```
Begin Transaction;
```

```
Insert into Client
```

```
values (10031, '00026', 'Bow Wow', '2016-08-12', 'Male', 'Dog', 17, 'Chihuahua Mix', 3)
```

```
commit;
```

```
/*Section A Milestone 5 Transaction #2
```

```
When adding a new groomer employee, also add the record to the 'child'
```

```
tables to reflect all of the employee's information. After Screenshot*/
```

```
Begin Transaction;
```

```
Insert into Employee
```

```
values (17, 'Tom', 'Brady', 65000_
```

```
Insert into Groomer
```

```
values (17, 'Michigan', '2018-08-07', 1)
```

```
Commit;
```

```
/*Section 2 Milestone 5 Trigger. Creates Trigger to increment SickDaysUSed in Employee
```

Table whenever an employee is unable to come to work. */

```
CREATE TRIGGER SickDayIncrement
    ON EmployeeSickDays
    After INSERT
AS
    UPDATE EMPLOYEE
        Set SickDaysUsed = (SickDaysUsed + 1)
        Where Employee# =
            (Select Employee#
             From inserted);
```

/* 2.(10) PROCEDURE: When a new boarding appointment is inserted, add a record to the BoardingLog and record ClientID, startdate, owner's name, and owners phone. Hint: You will need to create a BoardingLog to keep track of this data. The procedure will fire when a new boarding appointment is scheduled. */

--this is the create table statement

```
Create table BoardingLog (
    BoardingID int identity (1000, 5),
    Client# char(5),
    StartDate date,
    OwnerName varchar(20),
    OwnerPhone char(12))
```

--below is an example record to test our table

```
Insert into BoardingLog (Client#, StartDate, OwnerName, OwnerPhone)
    values ('10010', '2018-08-05', 'Garcia Lopez', '602-623-8754')
```

--this is the code we used to create the procedure, by referencing the client# to gather the owner information

```
Create Procedure AddBoardingLog
```

```
    @Client#          integer,
    @StartDate        date
AS
    Insert into BoardingLog (Client#, StartDate, OwnerName, OwnerPhone)
```

```

Values (
    @client#,
    @startdate,
    (select name
     from owner
     where owner# in (
         select owner#
         from client
         where client# = @client#)),
    (select phone
     from owner
     where owner# in (
         select owner#
         from client
         where client# = @client#)))

```

```
--this is a test transaction to verify that the owner (Jack Bauer) was listed as he should be in the
-- boardinglog table, upon execution of the procedure.
```

```
begin transaction;
```

```
exec addboardinglog 10005, '2019-09-08';
```

```
select * from boardinglog;
```

```
rollback;
```

```
commit;
```

```
/*Index advanced Join query used to tax the database. This query is designed to show
The Employee full name, salary count of that employee in appointments, sum of the taxes they paid,
The school that they attended for Grooming, The days that the client they helped had stayed,
The name of the client they helped. The number of days since they were certified */
```

```
set statistics time on;
```

```
Select distinct (E.EFName + E.ELName), E.Salary, count(A.employee#),
```

```

sum(PS.FederalTax + PS.StateTax) "Total Taxes", G.SchoolAttended, PS.BankName, count(BA.Client#),
Datediff (day, BA.CheckinDay, BA.CheckOutDay), C.Name, C.Client#,
DateDiff (Day, G.CertificateDate, GetDate()) "Days Since Certified"
From Employee E, Appointment A, Paystatement PS, Groomer G, Client C, BoardedAnimals BA,
ServicesProvided S
Where E.Employee# = A.Employee#
Group by E.EFName, E.ELName, E.Salary, G.SchoolAttended, PS.BankName, BA.CheckInDay,
BA.CheckOutDay,
G.CertificateDate, C.Name, PS.YTDSalary

```

/*Index advanced Join query used to tax the database. This query is designed to show
The Employee full name, salary count of that employee in appointments, sum of the taxes they paid,
The school that they attended for Grooming, The days that the client they helped had stayed,
The name of the client they helped. The number of days since they were certified Query
modified with Index inserted into the table */

```
set statistics time on;
```

```

Select distinct E.Salary, count(A.employee#),
sum(PS.FederalTax + PS.StateTax) "Total Taxes", G.SchoolAttended, PS.BankName, count(BA.Client#),
Datediff (day, BA.CheckinDay, BA.CheckOutDay),
DateDiff (Day, G.CertificateDate, GetDate()) "Days Since Certified"
From Employee E, Appointment A, Paystatement PS, Groomer G, Client C, BoardedAnimals BA,
ServicesProvided S
Where E.Employee# = A.Employee#
Group by E.EFName, E.ELName, E.Salary, G.SchoolAttended, PS.BankName, BA.CheckInDay,
BA.CheckOutDay,
G.CertificateDate, C.Name, PS.YTDSalary

```

```

/*Index Creation Statement for Client*/
Create Index ClientIndex on Client (Client#, Name)

```

```

/*Index creation statement for employee first name and last name */
Create index EmployeeIndex on Employee (EFName, ELName)

```

```
/*Creating a view and data visualization table to display how much  
each full-time employee has received in pay this year vs their annual salary */
```

```
Create view EmployeeYTDPay AS
```

```
    Select FT.Employee#, AccountNumber, YTDSalary, Salary  
        From Paystatement PS JOIN Employee E  
        On PS.Employee# = E.Employee#  
        Join FullTime FT  
        On FT.Employee# = E.Employee#
```

```
/*Creating a view and then Analytical Dashboard to show all of the number  
of days stayed by each client */
```

```
Create View TotalDaysStayed AS
```

```
    Select C.Name, O.Owner#, BA.ServiceCode,  
        DateDiff(day, BA.CheckINDay, BA.CheckOUTDay) "Number of Days Stayed"  
        From Client C Join Owner O  
        On C.Owner# = O.Owner#  
        Join BoardedAnimals BA  
        on BA.Client# = C.Client#
```

```
/*Creating a view describing the details of a given appointment, including relevant employee,  
client, and service info (including service fee)*/
```

```
Create view PetShopAppointments as
```

```
    Select appointment#, c.client#, e.employee#, EFName, ELName, Salary, Name, Gender,  
        Species, Weight, Sp.ServiceCode, ServiceType  
        From Appointment A  
        join client c  
        on c.client# = a.client#  
        join employee e  
        on e.employee# = a.employee#  
        join servicesprovided sp  
        on sp.servicecode = a.servicecode;
```

```
/*Creating a view showing each Breed, Species, and the number of services they had*/
```

```
create view mostservicesonclient as
    select Breed,Species, weight
        from CLIENT;
```

```
*****
*****
```

```
/*M2 Basic Queries*/
```

```
/*Names of Groomers, Their Clients, and Count of the Client */
```

```
Select E.EFName, E.ELName, C.Name, count(GA.Client#) "Count of Client"
    From Employee E Join Appointment A
        on E.Employee# = A.Employee#
    Join Client C
        on C.Client# = A.Client#
    Join GroomedAnimals GA
        on GA.Client# = A.Client#
    Group by E.EFName, E.ELName, C.Name
```

```
/*Names of Clients in the Last Year and Date they were Seen */
```

```
Select C.Name, BA.CheckINDay "Date Seen"
    From Client C Join BoardedAnimals BA
        on C.Client# = BA.Client#
    Where BA.CheckInDay >= '01-01-2019'
```

```
/*Services Provided Most Often by species */
```

```
Select C.Species, count(A.ServiceCode) "Count of Service Codes"
    From Client C Join Appointment A
        on C.Client# = A.Client#
    Group by C.Species
```

```
/*Services Provided for each client */
```

```
Select C.Name, count(A.ServiceCode) "Services Provided per Client"  
From Client C Join Appointment A  
on C.Client# = A.Client#  
Group by C.Name  
Order by count(A.ServiceCode)
```

```
/*List Each Client By DOB and Weight  
DOB was not recorded in our tables */
```

```
Select C.Name, C.Weight  
From Client C  
/*List all clients by Breed */  
Select C.Name, C.Breed  
From Client C  
Order by C.Breed  
/* For each animal boarded, list the client, the owner, the employee who was working  
and the number of days stayed */  
Select BA.Client#, C.Name, C.Owner#, O.Name,  
datediff(day, BA.CheckInDay, BA.CheckOutDay) "Number of Days Stayed"  
From BoardedAnimals BA Join Client C  
on BA.Client# = C.Client#  
Join Owner O  
on O.Owner# = C.Owner#
```

```
/* Which clients have been boarded in the last month? */
```

```
Select BA.Client#, C.Name, BA.CheckInDay  
From BoardedAnimals BA Join Client C  
on BA.Client# = C.Client#  
Where BA.CheckInDay >= '10/08/2019'
```

```
/*For all clients with a boarding appointment in the next week which do not have  
current vaccines */
```

```
Select BA.Client#, C.Name, BA.CheckInDay, G.CurrentVaccines
```

```
From BoardedAnimals BA Join Client C  
on G.Client# = C.Client#  
Where BA.CheckInDay >= '11/01/2019'
```

/*Client that has been boarded the most often */

```
Select BA.Client#, count(BA.Client#) "Number of Times Boarded"  
From BoardedAnimals BA  
Group by BA.Client#  
Order by count(BA.Client#) desc;
```

Which employees earn more than the average salary for all employees? */

```
Select E.Employee#, E.EFName, E.ELName, E.Salary, Avg(E.Salary) "Average Salary"  
From Employee E  
Group By E.Employee#, E.EFName, E.ELName, E.Salary  
Order By E.Salary
```

Appendix B:

Resumes

Baden Swallow:

PROFESSIONAL EXPERIENCE

Delivery Driver, Domino's Pizza, Inc., Tempe, AZ, Jan 2018-Current

- o Deliver goods to customers in a timely matter, ensuring satisfaction with products and encouraging return customers.
- o Completed transactions with customers both at the cash register and over the phone, speeding up the rate of service provided.
- o Interact with multiple managers and co-workers with different expectations over course of the day, allowing for smoother interactions with in-store staff and continued workflow.

Residential Assistant, Arizona State University Housing (ASU), Tempe, AZ, Jan 2019- May 2019

- o Established presence in residential community, fostering interpersonal development among residents.
- o Actively promote/conduct university events, supporting students' academic & professional interests throughout the academic school year.
- o Manage emergency crises & mediate resident conflicts, providing a more safe and secure residential hall for all occupants.
- o Assist with administrative tasks provided by University Housing, providing swift service and allowing for the continuing existence of residence hall

SKILLS, AWARDS, ACTIVITIES

Technical skills: Microsoft Office: Word, Excel, Power Point, Access. Java, SQL

Robert Nelson

9940 Newton Court | Westminster, CO 80031 | (720) 412-5546 | robert920nelson@gmail.com

I'm a student in the Leader's Academy at W. P. Carey School of Business at Arizona State University studying Business Data Analytics, with a second major in Music Theater. I enjoy working with business data analysis software and have earned a 4.01 GPA my first semester. Outside of school I have practiced with Tableau. I am eager to continue learning more ways to process, assess, and report information and to contribute in a real business environment.

Skills

Microsoft Excel, Word, Access

Microsoft Azure

Tableau

Google Docs

SQL

Multivariate Calculus (5 on AP B/C exam)

College Coursework:

- Accounting
- Economics
- Statistics

Education

Business Data Analytics, BS

Arizona State University, Tempe, AZ

W. P. Carey School of Business, Leader's Academy

May 2021

GPA: 4.01

International Baccalaureate Diploma

Standley Lake High School, Westminster, CO

May 2017

Weighted GPA: 4.72

Class Rank: 4 of 257

Volunteer Experience

President of Key Club

Standley Lake High School, Westminster, CO

May 2016 - May 2017

- Organized events by contacting leaders in various programs
- Raised roughly \$2,000 in group funds to be donated to local charities

National Honor Society Member

Standley Lake High School, Westminster, CO

May 2015 - May 2017

- Participated in group projects to raise funds for charities by showing leadership and initiative within school and community environments
- 20+ hours of community service annually

Awards

Fall 2017 Dean's List*Arizona State University, Tempe, AZ*

January 2018

- Maintained a 3.5 or higher GPA while enrolled in 12 or more credit hours

Standley Lake Honor Roll*Standley Lake High School, Westminster, CO*

2014 - 2017

- Maintained a 3.8 or higher GPA

Bryan Flynn

6146 W. Questa Dr., Glendale, Arizona 85310

(623) 810-3130 bhflynn@asu.edu

Professional Summary

Multi-talented professional consistently rewarded for success in planning and operational improvements. Experience in policy development and staff management procedures positively impacting overall morale and productivity.

Skills

- **Microsoft Office**
- **SQL**
- **Data Analytics Programs**
- **Proficient in Mathematics as excelled in calculus**

Work History

Asset Protection Specialist | 01/2017 to Current

Fry's Food And Drug Stores - Glendale, Arizona

- Increased company understanding of loss control strategies through on-site client training seminars and implementation of new programs
- Introduced and implanted viable loss control strategies by maintaining strong partnerships and trust with customers
- Enhanced training processes for both clients and departments to increase overall functionality
- Collected and organized all surveillance data and information to protect client possessions and workspaces

Education

Arizona State University - Tempe, AZ | Bachelor of Science

Business Data Analytics, Expected in 06/2020

Certifications

Wiklander-Zulawski certified used for professional interviews of associates within companies.

NAYAN TEZ

977 E Apache Blvd. Tempe AZ

E-mail: 9tez99@gmail.com/9tez@asu.edu

Phone: +14807400505

OBJECTIVE: To pursue an internship in the supply chain management industry and get equipped with knowledge and expertise, gain global exposure that is required to develop my career in the field of business while continuing my undergraduate study at W. P. Carey School of Business.

EDUCATION			
Year	Degree	Institute/School	% / CGPA
2017-2020	Supply-chain Management& Data analytics	W. P. Carey School of Business Arizona State University	3.42/4.0

Work/Projects

- Interned at Synergies Casting Limited, Visakhapatnam in the Supply Chain Department.
 - Interned at Blockchain Enterprise, a supply chain startup that uses block chain technology to track and improve

Supply-chain visibility for its clients where I was responsible for creating presentations

for potential investors.

2018

- Working on a business plan for a unique small-scale project in the hospitality sector in Visakhapatnam and awaiting approval from VUDA for approvals.

2018

Leadership Experience:

- School swimming team captain 2012-2014
- ASU Squash Team 2017

Extracurricular activities:

- State level swimmer. Participated in numerous championships. 2012-2014
- Regular squash player in the past 3 years. 2013
- Enjoy all types of adventure activities such as Scuba diving, trekking, rafting, camping, and mountaineering.
- Participated in many Cycling brevets and marathons such as the ‘Olympic Day Run 2013’ held at Visakhapatnam contributing to the Guinness World Record. 2012
- Organized college fêtes successfully in the years 2015 &2016.
- Made a few short films with social messages.

Community services:

- Pep rallies to promote blood donations, breast cancer awareness and various other social causes 2012
- Supported Girl child education through a nonprofit organization (*Edupuganti foundation*) 2014
- Community cleanup program (*Swachh Bharath Program*) 2015
- Disaster relief works for Chennai floods in 2013 and a major cyclone in Vizag (*HUD HUD*) in the year 2014.

Personal details:

- Nationality: Indian
- Languages Known: English, Telugu, Hindi, Sanskrit and French.
 - Hobbies: Squash, Swimming, Gourmet Cooking, Browsing Internet, Watching Movies & TV shows.
 - Skills: Proficient in Microsoft office suite, Adobe Suite and other basic SCM applications like SQL programming.

Team Member Name:	Baden Swallow	
Date of Work Performed	Description of Work Performed	Time Spent on Work Performed (In Hours)
8/29/2019	Slack Setup/Development	2
9/10/2019	Team meeting #1 and divided tasks to be completed	0.75
9/10/2019	Developed intro paragraph for Nayan	0.42
9/15/2019	Input list of skills and resume into document	0.33
9/15/2019	Formatting of Final Document	0.25
9/15/2019	Proof milestone before submitting	0.25
9/24/2019	Team meeting going over milestone 2 and assigning parts	1
9/26/2019	Edited and modified existing ERD to fit given guidelines.	1
9/26/2019	Added additional entries to the data dictionary.	0.25
9/29/2019	Proofed final document for submission.	0.5
10/20/2019	Initial creation/start of sheet used for insert query.	0.5
10/27/2019	Continued to fill in insert data to sheet.	3
10/27/2019	Continued to work on TXT file	1.5
11/7/2019	Team Meeting #4 and delegated tasks required for Milestone 4	1
11/10/2019	Created queries to test database system for Milestone 4	6
11/10/2019	Proof milestone before submitting	0.25
11/23/2019	Began work on constructing trigger/procedure statements for M5	2.5

11/26/2019	Team Meeting to work on Milestone 5	1
11/26/2019	Created Analytics Dashboard and constructed Google Doc	2
12/1/2019	Began M7 doc and compiling of necessary information	1
12/4/2019	Construction of project presentation	0.5
12/8/2019	Completion of M7 doc with Bryan & Robert	3
	Total Hours for: Baden Swallow	29

Team Member Name:	Bryan Flynn	
Date of Work Performed	Description of Work Performed	Time Spent on Work Performed (In Hours)
9/10/2019	Developed intro paragraph for Baden	0.5
9/10/2019	Team meeting and divided tasks to be completed	1
9/11/2019	Completed bullet list compiling skills, technology, and assigned role info.	1.5
9/13/2019	Adjusted resume to current date	0.5
9/15/2019	Added resumes for both robert and I to final document	0.25
9/15/2019	Added my assigned sections to the final document and proofed doc	0.5
9/24/2019	Team meeting going over milestone 2 and assigning parts	1
9/26/2019	Began Data dictionary and added some mock data	0.5
9/26/2019	Worked on ERD creating entities and cardinalities	1
9/29/2019	Fixed Journals to show time in hours and not in minutes	0.25
9/29/2019	Proofing ERD as well as data dictionary to ensure accuracy	0.75
10/26/2019	Updated Data dictionary to reflect updated ERD	1
10/26/2019	Created Create table TXT file	2
10/27/2019	Finished ERD, filled in some insert data, and finished milestone	4
11/8/2019	Team meeting regarding milestone 4	1
11/10/2019	Created 2 advanced scripts 1 view and 1 group by	2
11/10/2019	Proof milestone before submitting	0.25

11/21/2019	Created Transaction SQL Queries	1
11/26/2019	Team meeting to work on M5	3
11/26/2019	Completed Index Tables and completed IC-11 for team	3
11/26/2019	Finalized text file and google doc	2
12/1/2019	Began M7 doc and compiling of necessary information	3
12/4/2019	Construction of project presentation	1
12/8/2019	Completion of M7 doc with Baden & Robert	4
	Total Hours for: Bryan Flynn	31

Team Member Name:	Robert Nelson	
Date of Work Performed	Description of Work Performed	Time Spent on Work Performed (In Hours)
9/9/2019	Organization/Planning of Responsibilities	1
9/10/2019	Team Meeting #1, discussed responsibilities for each member	1
9/10/2019	Compiled work in singular document, formatted paper	1.33
9/13/2019	Title page, wrote James' paragraph	0.75
9/15/2019	Final review of paper, uploaded headshot	0.17
9/24/2019	Team meeting going over milestone 2 and assigning parts	1
9/24/2019	Developed skeleton of ERD for group	2
9/29/2019	Proofed final document for accuracy and completeness	0.5
10/24/2019	Meeting, discussed responsibilities for insert statements	1
10/26/2019	Structured insert statements in google sheets	4
10/27/2019	Troubleshoot inserts and final load	2
11/3/2019	In class meeting, discussed initial responsibilities	0.5
11/7/2019	Team meeting, discuss responsibilities and methods for editing tables	1
11/10/2019	Basic query work, reformatting and reapplying table information	4

11/10/2019	Proof milestone before submitting	0.25
11/25/2019	Created Visual Analysis in Tableau	2
11/26/2019	Team meeting discussing and working on M5	3
11/26/2019	Created Procedure for final M5 doc	1
12/4/2019	Construction of project presentation	0.5
12/8/2019	Appendix A Completion, histogram, and formatting of A7	2
	Total Hours for: Robert Nelson	29

Team Member Name:	Nayan Tez Vankayalapati	
Date of Work Performed	Description of Work Performed	Time Spent on Work Performed (In Hours)
9/15/2019	Updated resume	0.33
9/15/2019	Proofread document before submission	0.25
9/24/2019	Team meeting going over milestone 2 and assigning parts	1
9/27/2019	Created group gliffy workspace	0.6
9/27/2019	Developed ERD, added attributes	1
9/28/2019	Finished adding attributes	1.5
9/29/2019	Proofread milestone#2 and the ERD	1.2
10/24/2019	Created and completed Database schema	3
10/27/2019	Contributed to data sheet to complete TXT file	3
11/10/2019	Created advanced nested queries for Milestone 4	5
11/10/2019	Proof milestone before submitting	0.25
	Total Hours for: Nayan Tez Vankayalapati	17.13

Appendix C:

Team Meeting Agendas

Date & Time: ____ 09/10/2019 ____

Team Meeting for: Team 3 (Jurassic Park)

Prepared by: *BGS*

Client's Name: TBA

Team members in attendance:

1. (Chair) - Baden
2. James
3. Bryan
4. Robert

Team members absent:

1. Nayan

Meeting Objective:

Introduced all present team members, discussed project requirements, and allocated individual work for each section of Milestone 1.

Action Item	Assigned To:	Due Date:
Formal Team Introduction	All Present	09/10/2019
Team Image/Code of Ethics	Primary- Robert (Split among entire team)	09/15/2019
Team Profile	Every team member assigned individually	09/15/2019
Development of communication platform & cloud sharing	Primary- Baden (Develop as needed by team)	09/12/2019

Time meeting ended: _____ 8:45 AM _____

Date and time of next meeting:

_____ 09/19/2019 at 8:00 AM _____

Date & Time: _____ 09/24/2019 _____

Team Meeting for: Team 3 (Jurassic Park)

Prepared by: BHF

Client's Name: Dip and Clip

Team members in attendance:

1. (Chair) - Robert
2. James
3. Bryan
4. Baden
5. Nayan

Team members absent:

1. N/A

Meeting Objective:

Discussed milestone 2 and went over the company information that our project is on.
Also establish a game plan for completing this milestone.

Action Item	Assigned To:	Due Date:
Milestone 2 Intro	All Present	
Divided items needed for ERD	Primary- Robert (Split among entire team)	09/27/2019
Data Dictionary	Primary- James	09/27/2019
Established better ways to communicate and additional meeting time	Entire Team	09/29/2019

Time meeting ended: _____ 8:55 AM _____

Date and time of next meeting:

_____TBD_____

Date & Time: _____ 11/07/2019 _____

Team Meeting for: Jurassic Park Team 3

Prepared by: Bryan Flynn

Client's Name: Dip And Clip Dog Grooming

Team members in attendance:

1. Robert
2. Bryan
3. Baden

- 4.
- 5.

Team members absent:

1. Nayan

Meeting Objective:

Establish time frame for milestone 4 and delegate how and when each other's SQL codes are due.

Action Item	Assigned To:	Due Date:
Basic Queries/Advanced	Robert	11/09/2019
Basic Queries/Advanced	Bryan	11/09/2019
Basic Queries/Advanced	Nayan	11/09/2019
Basic Queries/Advanced	Baden	11/09/2019

Time meeting ended: ____ 9:00AM _____

Date and time of next meeting:

____ 11/12/19 ____

Date & Time: ____ 11/12/2019 _____

Team Meeting for: Team 3 (Jurassic Park)

Prepared by: Bryan Flynn

Client's Name: Dip & Clip

Team members in attendance:

1. (Chair) - Baden

2. Nayan
3. Bryan
4. Robert

Team members absent:

Meeting Objective:

This Meeting Was used to review in class lecture on optimization/indexes and begin discussing our plans for how to complete this assignment

Action Item	Assigned To:	Due Date:
Review Indexes and Optimization	All Present	11/17/2019
Indexes were given to team members to be completed	Primary- Nayan/Robert	11/17/2019
Optimizations were to be handled	Baden and Bryan	11/17/19

Time meeting ended: _____ 10:15 AM _____

Date and time of next meeting:

_____ 11/14/2019 at 9:00 AM _____

Date & Time: _____ 12/08/2019 _____

Team Meeting for: Team 3 (Jurassic Park)

Prepared by: *Bryan Flynn*

Client's Name: Dip & Clip

Team members in attendance:

1. (Chair) - Baden
3. Bryan

Team members absent:

1. Nayan
2. Robert

Meeting Objective:

This meeting was used to complete M7 and all that it entails.

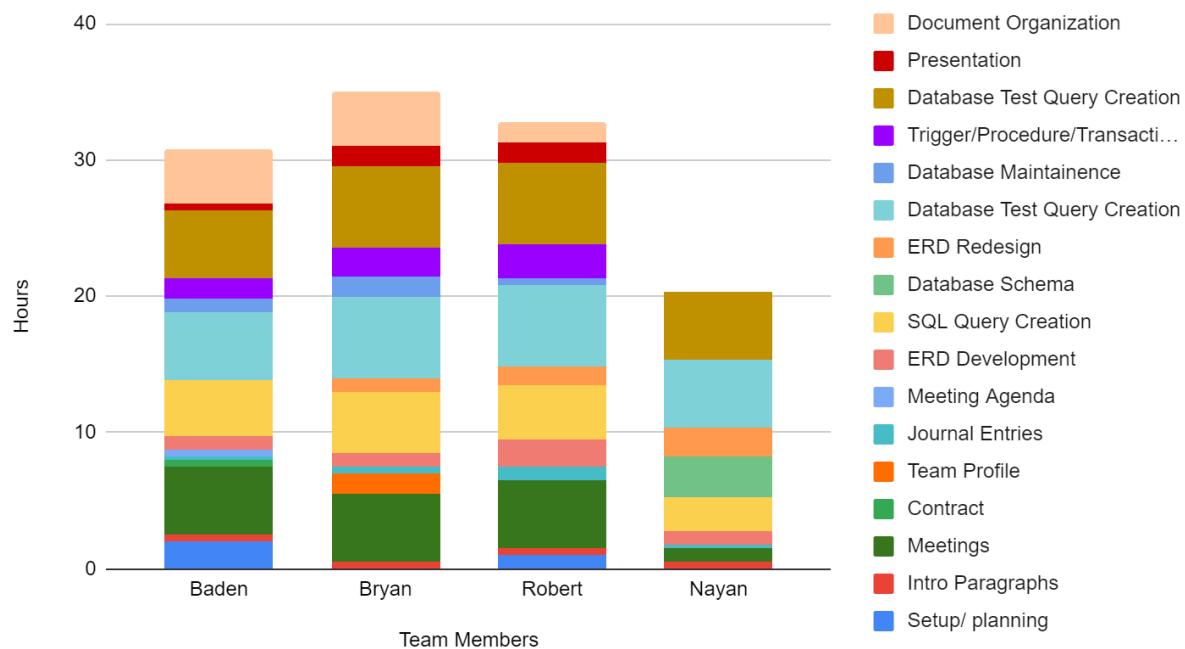
Action Item	Assigned To:	Due Date:
Compiling all information into M7 doc	Baden and Bryan	12/08/2019

Time meeting ended: _____ 9:00 PM _____

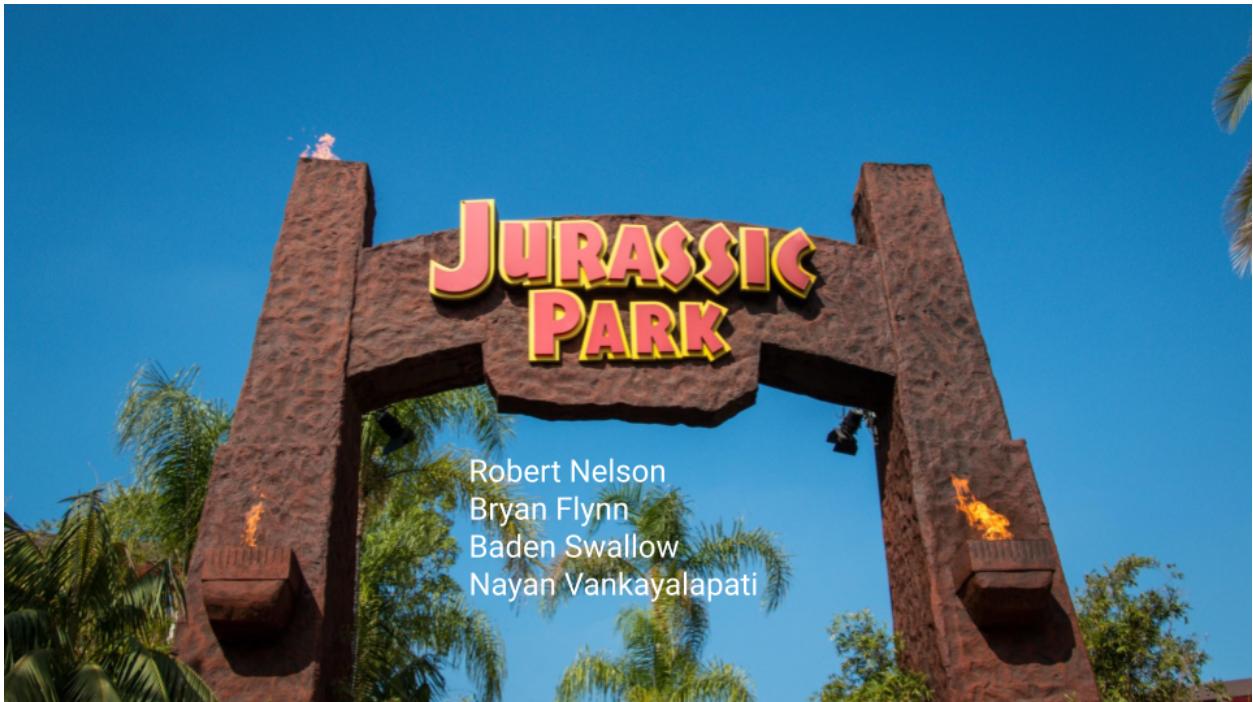
Date and time of next meeting:

____ N/A _____

Team Jurassic Park Resource Histogram



Appendix D



Meet our team:

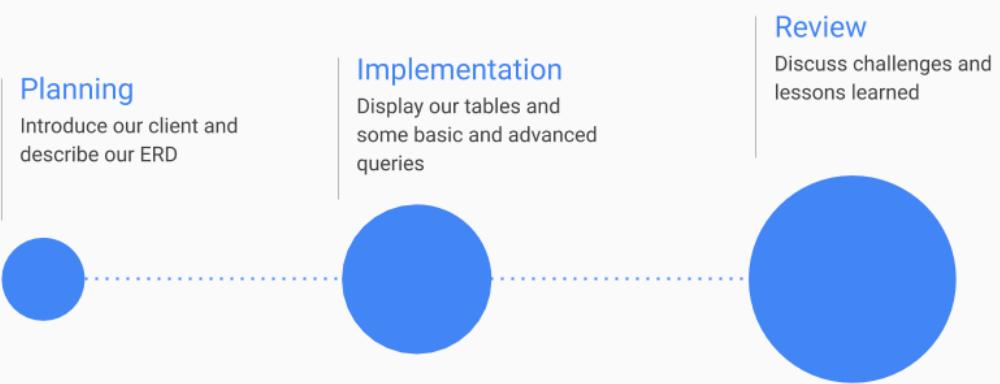
Bryan Flynn
Conatation: Quick Start

Baden Swallow
Conatation: Facilitator

Robert Nelson
Conatation: Follow Through

Nayan Tez Vankayalapati
Conatation: Follow Through

Our roadmap:

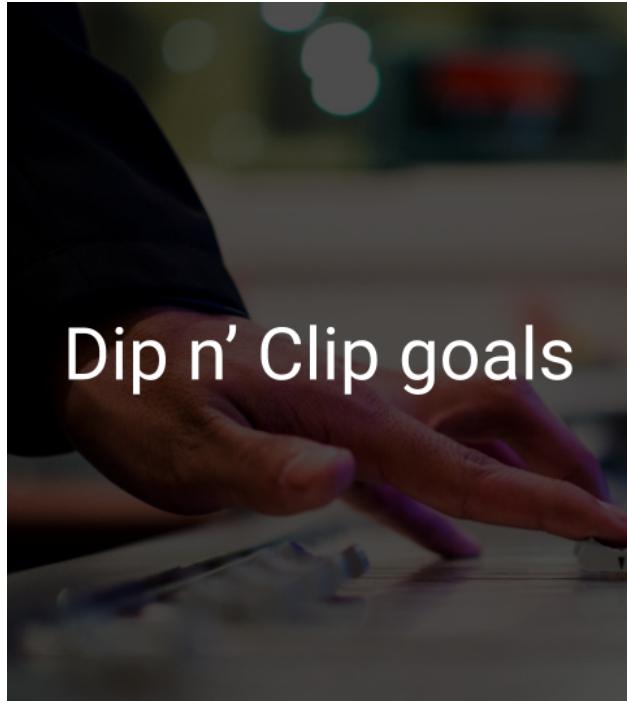


Dip and Clip

Grooming Salon and Boarding Kennel

- Pet Grooming and Boarding
- Located in Flagstaff, AZ
- Sharon & Jody Prescott
- 5 years
- 12 Full Time Employees, 4 Part Time





Dip n' Clip goals

- Get the record keeping up to date.
- Offer more services to their clients.

Our Client's Data

<u>Weight of animal</u>	<u>Short Hair</u>	<u>Long Hair</u>
5-20lbs	\$20	\$25
21-40lbs	\$30	\$45
41-60lbs	\$40	\$55
61-80lbs	\$50	\$65
81+ lbs.	\$60	\$70

Extra charge for matted hair \$30 per animal

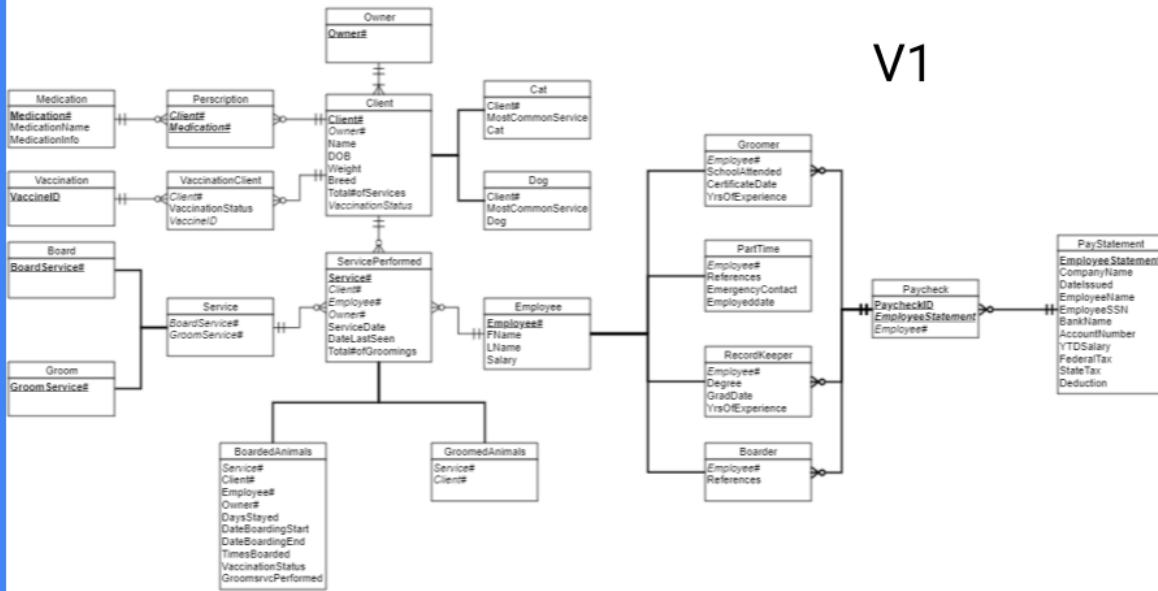
Service Code	Service Description	Fee
1.0	Walk in Services	
1.01	nail trim	8.00
1.02	ear cleaning	5.00
1.03	skin scrape	25.00
1.04	ear mite treatment	10.00
1.05	flea treatment	15.00
1.06	tick treatment	12.50
1.07	teeth brushing	8.50
1.08	nose & pad conditioning	6.00
1.09	eye cleaning & drops	7.50
1.10	hair cut	15.00
1.11	chalking(5 colors available)	4.00
1.12	stenciling (20 designs, 8 color options)	7.00
1.13	feathering (6 options available)	5.00
1.14	Vitamins (soft shiny coat)	15.00
1.15	other (specify)	
2.0	Boarding Services	
2.01	medication applications (3+)	10.00
	matted hair	30.00

Our Client's Excel Spreadsheet

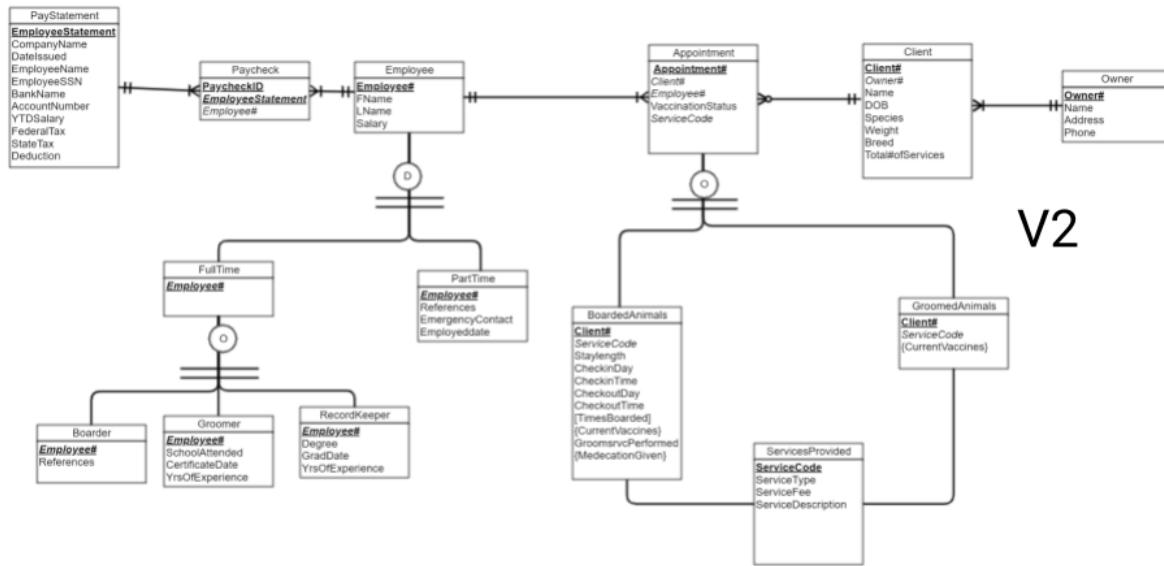
ERD Process

Dip & Clip Grooming Salon and Boarding

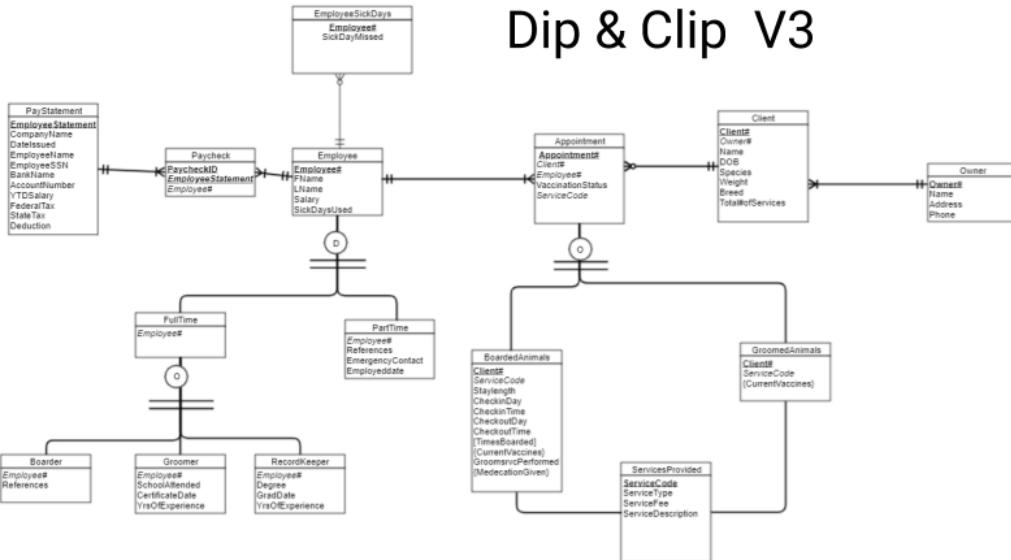
V1



V2



Dip & Clip V3



select *
from employee

	Employee#	EFName	ELName	Salary
1	01	Jonathan	Selmy	42450.00
2	02	Amanda	Lath	51000.00
3	03	Sam	Forwell	46000.00
4	04	Riley	Shorwich	23000.00
5	05	Dale	Timmel	54000.00
6	06	Olivia	Alaway	25000.00
7	07	Abigail	Chavez	88000.00
8	08	Timothy	Lowell	67000.00
9	09	Ross	Anders	32000.00
10	1	Jonathan	Selmy	42450.00
11	10	Grace	Robertson	56000.00
12	11	Courtney	Alcine	43000.00
13	12	Paul	Elworth	44000.00
14	2	Amanda	Lath	51000.00
15	3	Sam	Forwell	46000.00
16	4	Riley	Shorwich	23000.00
17	5	Dale	Timmel	54000.00
18	6	Olivia	Alaway	25000.00
19	7	Abigail	Chavez	88000.00
20	8	Timothy	Lowell	67000.00
21	9	Ross	Anders	32000.00

select *
from client

	Client#	Owner#	Name	DOB	Species	Weight	Breed
1	10001	00002	Sparky	2014-03-12	Dog	40	Golden Retriever
2	10002	00019	Bubbles	2010-01-02	Dog	22	Basset Hound
3	10003	00005	Jazz	2015-11-23	Cat	14	Burmese
4	10004	00011	Lucky	2012-11-14	Cat	10	Shorthair
5	10005	00001	Toby	2008-06-29	Dog	12	Chinese Crested Dog
6	10006	00029	Cal	2017-07-19	Cat	15	Bengal
7	10007	00008	Rotor	2014-03-06	Dog	68	Great Dane
8	10008	00013	Tyson	2016-07-15	Cat	15	Sphynx
9	10009	00007	Scrabi	2015-01-26	Dog	26	Welsh Corgi
10	10010	00017	Tiddie	2015-06-11	Dog	38	Chocolate Labrador
11	10011	00026	Fabio	2011-09-09	Cat	12	Persian
12	10013	00023	Maggie	2018-02-14	Cat	16	Siamese
13	10014	00028	Buttons	2017-05-19	Cat	10	Chartreux
14	10015	00030	Chad	2015-07-05	Cat	11	Turkish Van
15	10016	00006	Sprt	2017-08-13	Cat	15	Manx
16	10017	00014	Clover	2009-12-25	Cat	14	Burmese
17	10018	00003	Anna...	2012-12-07	Dog	22	Jack Russel Terrier
18	10019	00025	Lari	2013-06-04	Cat	18	Japanese Bobtail
19	10020	00016	Whisk...	2016-04-22	Cat	14	Sphynx
20	10021	00009	Rouge	2016-03-09	Dog	46	Golden Retriever
21	10022	00022	Chester	2010-01-13	Cat	20	Birman
22	10023	00027	Buck	2017-01-29	Cat	14	Korat
23	10024	00010	Chewie	2016-10-17	Dog	13	Pomeranian
24	10025	00024	Puppin	2011-08-14	Dog	19	English Mastiff
25	10026	00020	Pebbles	2012-07-27	Dog	20	Rottweiler
26	10027	00012	Vienne	2016-04-20	Dog	22	Boxer
27	10028	00021	Troy	2015-04-12	Cat	16	Turkish Van



Which **employees** attended ASU?



**Arizona State
University**

Results

	Employee#	paycheckID
1	07	13467
2	11	21862
3	12	41623
4	07	43860
5	11	60213
6	11	66323
7	07	74554
8	07	74736
9	12	98945

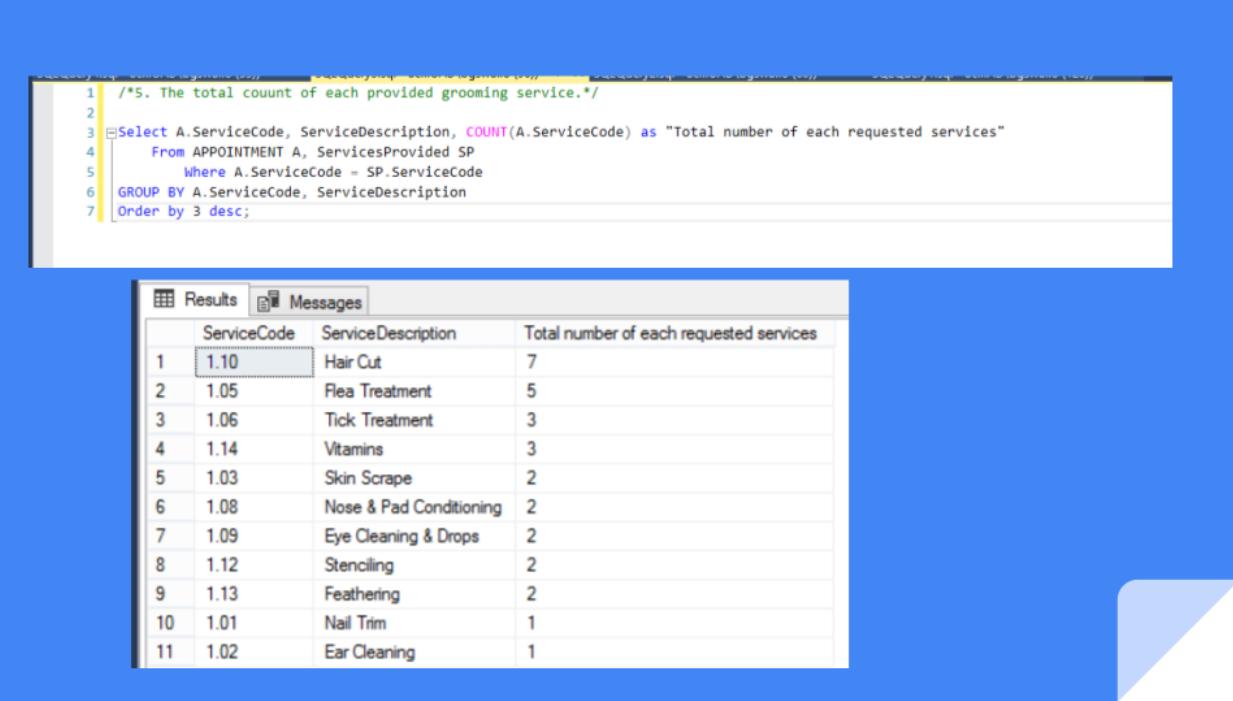
SQLQuery3.sql - ac...UAD\bgswallo (96)* → X SQLQuery2.sql - ac...UAD\bgswallo (60)* → SQLQuery1.sql - ac...UAD\bgswallo (60)*

```
1 /*5. The PaycheckID for workers that went to ASU */
2 Select Employee#, paycheckID
3   From PAYCHECK
4     Where employee# in
5       (Select employee#
6         From EMPLOYEE
7           Where employee# in
8             (Select employee#
9               From GROOMER
10              Where schoolattended ='ASU'));
```



Which services are **most** popular?

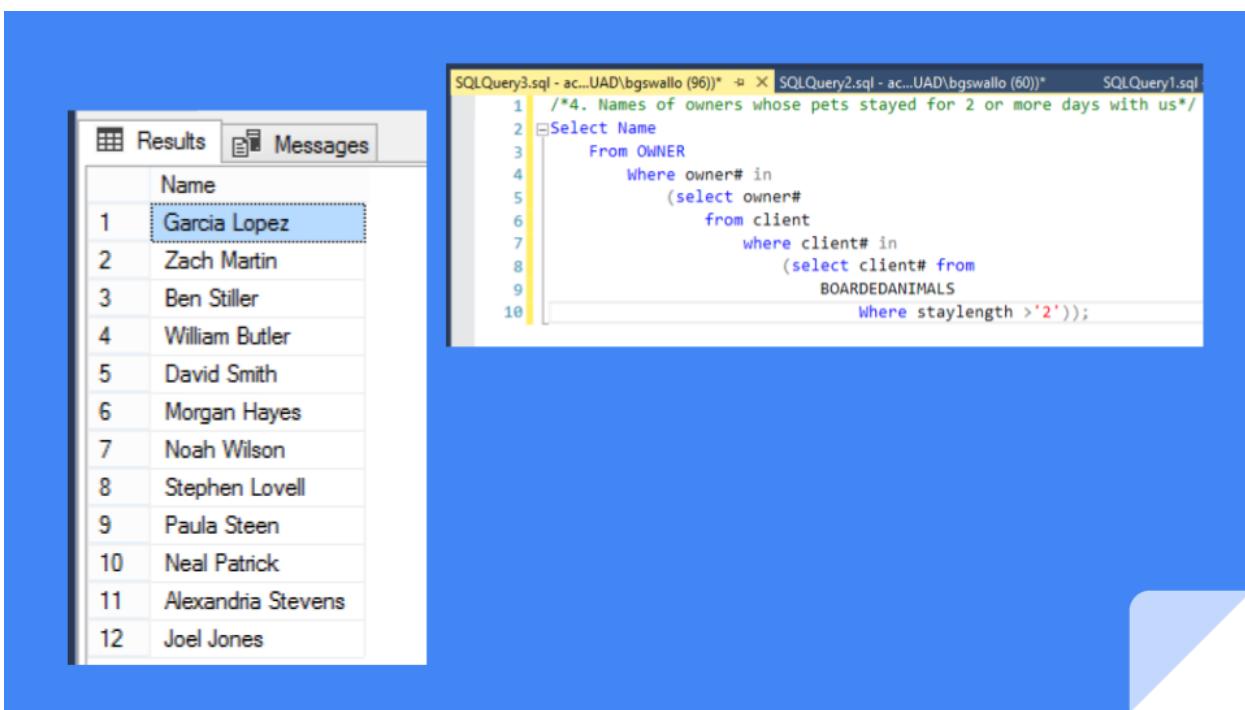
```
1  /*5. The total count of each provided grooming service*/
2
3  Select A.ServiceCode, ServiceDescription, COUNT(A.ServiceCode) as "Total number of each requested services"
4    From APPOINTMENT A, ServicesProvided SP
5      Where A.ServiceCode = SP.ServiceCode
6    GROUP BY A.ServiceCode, ServiceDescription
7    Order by 3 desc;
```



	ServiceCode	ServiceDescription	Total number of each requested services
1	1.10	Hair Cut	7
2	1.05	Flea Treatment	5
3	1.06	Tick Treatment	3
4	1.14	Vitamins	3
5	1.03	Skin Scrape	2
6	1.08	Nose & Pad Conditioning	2
7	1.09	Eye Cleaning & Drops	2
8	1.12	Stenciling	2
9	1.13	Feathering	2
10	1.01	Nail Trim	1
11	1.02	Ear Cleaning	1

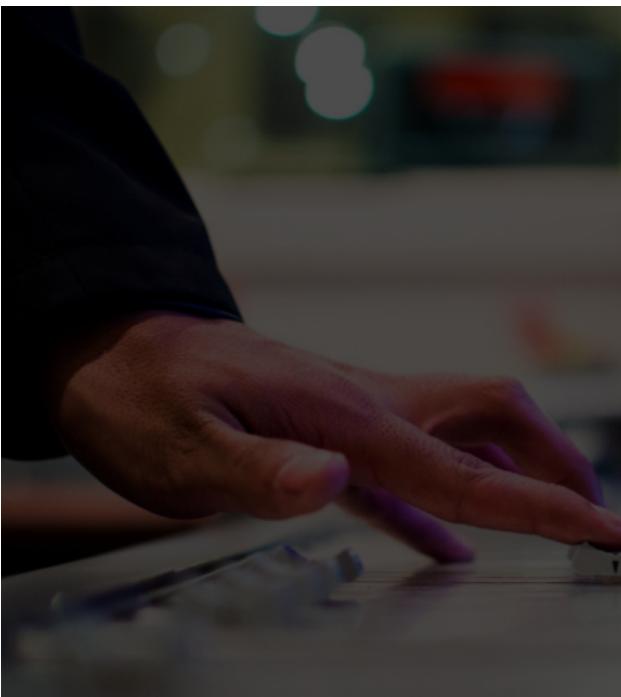


Which owners boarded their pets for **longer than 2 days?**



	Name
1	Garcia Lopez
2	Zach Martin
3	Ben Stiller
4	William Butler
5	David Smith
6	Morgan Hayes
7	Noah Wilson
8	Stephen Lovell
9	Paula Steen
10	Neal Patrick
11	Alexandria Stevens
12	Joel Jones

```
SQLQuery3.sql - ac...UAD\bgswallo (96)* X SQLQuery2.sql - ac...UAD\bgswallo (60)* SQLQuery1.sql
1 /*4. Names of owners whose pets stayed for 2 or more days with us*/
2 Select Name
3   From OWNER
4     Where owner# in
5       (select owner#
6         from client
7           where client# in
8             (select client# from
9                 BOARDEDANIMALS
10                Where staylength >'2'));
```



Advanced Queries

```
SQLQuery4.sql - ac...UAD\bgswallo (55)*      SQLQuery3.sql - ac...UAD\bgswallo (96)*  ✘  SQLQuery2.sql - ac...UAD\bgswallo (60)*      SQLQuery1.sql - ac...AD\bgswallo (120)*
1  /*5. The total count of each provided grooming service.*/
2
3  Select A.ServiceCode, ServiceDescription, COUNT(A.ServiceCode) as "Total number of each requested services"
4    From APPOINTMENT A, ServicesProvided SP
5       Where A.ServiceCode = SP.ServiceCode
6   GROUP BY A.ServiceCode, ServiceDescription
7   Order by 3 desc;
```

Results

ServiceCode	ServiceDescription	Total number of each requested services
1.10	Hair Cut	7
1.05	Rea Treatment	5
1.06	Tick Treatment	3
1.14	Vitamins	3
1.03	Skin Scrape	2
1.08	Nose & Pad Conditioning	2
1.09	Eye Cleaning & Drops	2
1.12	Stenciling	2
1.13	Feathering	2
1.01	Nail Trim	1
1.02	Ear Cleaning	1

SQLQuery3.sql - ac...UAD\bgswallo (99)* → SQLQuery2.sql - ac...UAD\bgswallo (60)* → SQLQuery1.sql - ac...AD\bgswallo (120)*

```

1  /*5. The PaycheckID for workers that went to ASU */
2  Select Employee#, paycheckID
3    From PAYCHECK
4      Where employee# in
5          (Select employee#
6            From EMPLOYEE
7              Where employee# in
8                  (Select employee#
9                    from GROOMER
10                   Where schoolattended = 'ASU'));

```

110 %

Results Messages

Employee#	paycheckID
07	13467
11	21862
12	41623
07	43860
11	60213
11	66323
07	74554
07	74736
12	98945

SQLQuery1.sql - ac...AD\bgswallo (120)* → SQLQuery2.sql - ac...UAD\bgswallo (80)*

```

1  /* Section 2 Milestone 5 Trigger. Creates trigger to increment
2   "sickdaysused" in EMPLOYEE table whenever an employee is unable
3   to come to work.*/
4
5  CREATE TRIGGER SickDayIncrement
6    ON EmployeeSickDays
7      After INSERT
8
9    AS
10   UPDATE EMPLOYEE
11     Set SickDaysUsed = (SickDaysUsed + 1)
12       Where Employee# =
13           (Select Employee#
14             From inserted);

```

5 BEGIN TRANSACTION

6 Select * From EMPLOYEE;

7 Insert Into EmployeeSickDays Values('02', '1');

8 Select * From EMPLOYEE;

9 --Rollback;

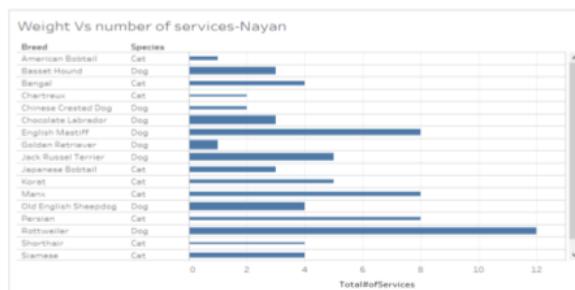
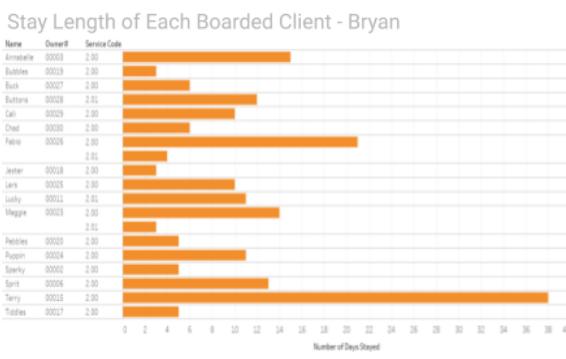
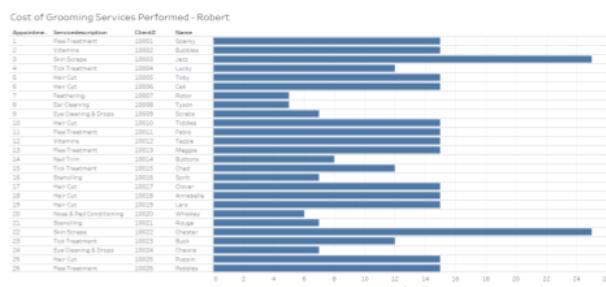
120 %

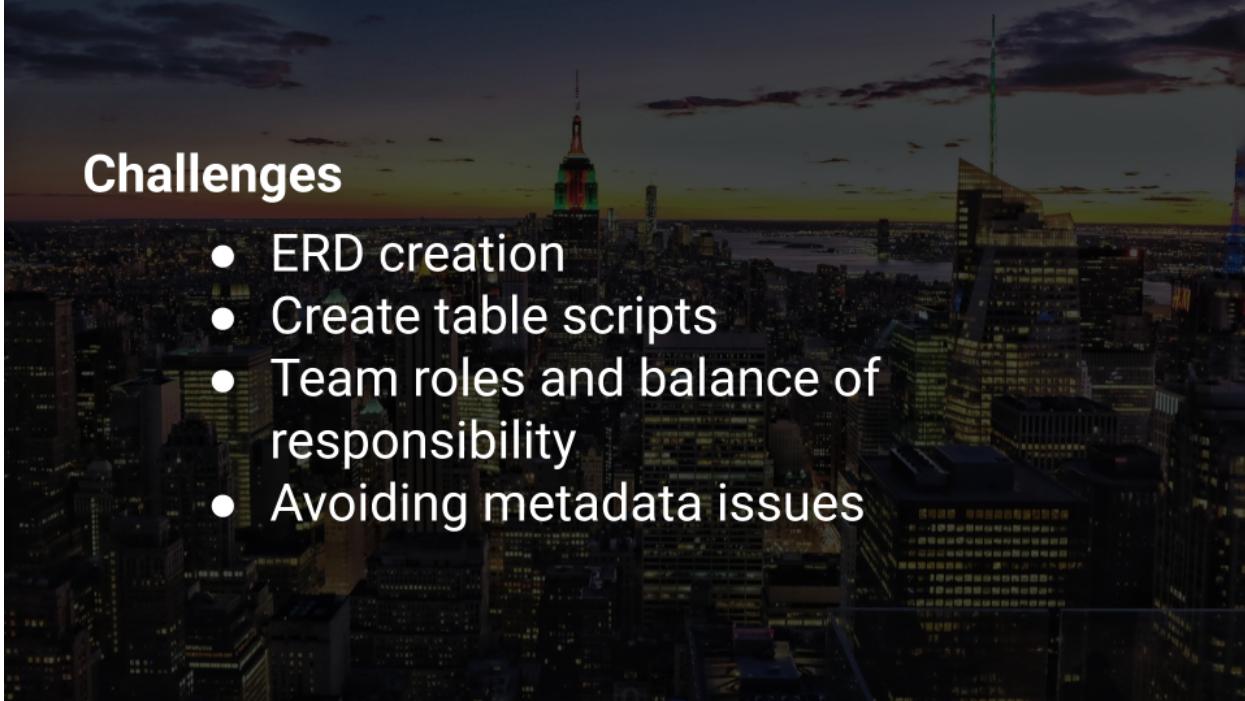
Results Messages

Employee#	EFName	ELName	Salary	SickDaysUsed
01	Jonathan	Selmy	42450.00	2
02	Amanda	Lath	51000.00	0
03	Sam	Forwell	46000.00	2
04	Riley	Shorwich	23000.00	2
05	Dale	Timmel	54000.00	0
06	Olivia	Alaway	25000.00	0
07	Abigail	Chavez	88000.00	0
08	Timothy	Lowell	67000.00	0

Employee#	EFName	ELName	Salary	SickDaysUsed
01	Jonathan	Selmy	42450.00	2
02	Amanda	Lath	51000.00	1
03	Sam	Forwell	46000.00	2
04	Riley	Shorwich	23000.00	2
05	Dale	Timmel	54000.00	0

```
1 /* 2.10 PROCEDURE: When a new boarding appointment inserted, add a record to the BoardingLog and record ClientID, 
2 startdate, owner's name, and owners phone. Hint: You will need to create a BoardingLog to keep track of this data.
3 The procedure will fire when a new boarding appointment is scheduled. */
4 
5 --this is the create table statement
6 Create table BoardingLog (
7     BoardingID int identity (1000, 5),
8     Client char(5),
9     StartDate date,
10    OwnerName varchar(20),
11    OwnerPhone char(12));
12 
13 --below is an example record to test our table
14 Insert into BoardingLog (Client#, StartDate, OwnerName, OwnerPhone)
15     values ('10000', '2018-08-05', 'Garcia Lopez', '602-623-8754')
16 
17 
18 --this is the code we used to create the procedure, by referencing the client# to gather the owner information
19 Create_Procedure AddBoardingLog
20     @Client# integer,
21     @StartDate date
22 
23 AS
24     Insert into BoardingLog (Client#, StartDate, OwnerName, OwnerPhone)
25         Values (
26             @Client#,
27             @StartDate,
28             (select name
29                 from owner
30                 where owner in (
31                     select owner#
32                         from client
33                         where client# = @Client#)),
34             (select name
35                 from owner
36                 where owner in (
37                     select owner#
38                         from client
39                         where client# = @Client#)));
40 
41 --this is a test transaction to verify that the owner (Jack Bauer) was listed as he should be in the
42 -- boardinglog table, upon execution of the procedure.
43 
44 begin transaction;
45 exec addboardinglog 10000, '2019-09-05';
46 select * from boardinglog;
47 
48 rollback;
49 |commit;|
50 % - 4
51 Results Messages
52 
53 BoardingID Client# StartDate OwnerName OwnerPhone
54 1 10000 2018-08-05 Garcia Lopez 602-623-8754
55 2 10005 2019-09-08 Jack Bauer 602-555-0170
56 
57 
58 
59 
60 
61 
62 
63 
64 
65 
66 
67 
68 
69 
70 
71 
72 
73 
74 
75 
76 
77 
78 
79 
80 
81 
82 
83 
84 
85 
86 
87 
88 
89 
90 
91 
92 
93 
94 
95 
96 
97 
98 
99 
100 
101 
102 
103 
104 
105 
106 
107 
108 
109 
110 
111 
112 
113 
114 
115 
116 
117 
118 
119 
120 
121 
122 
123 
124 
125 
126 
127 
128 
129 
130 
131 
132 
133 
134 
135 
136 
137 
138 
139 
140 
141 
142 
143 
144 
145 
146 
147 
148 
149 
150 
151 
152 
153 
154 
155 
156 
157 
158 
159 
160 
161 
162 
163 
164 
165 
166 
167 
168 
169 
170 
171 
172 
173 
174 
175 
176 
177 
178 
179 
180 
181 
182 
183 
184 
185 
186 
187 
188 
189 
190 
191 
192 
193 
194 
195 
196 
197 
198 
199 
200 
201 
202 
203 
204 
205 
206 
207 
208 
209 
210 
211 
212 
213 
214 
215 
216 
217 
218 
219 
220 
221 
222 
223 
224 
225 
226 
227 
228 
229 
230 
231 
232 
233 
234 
235 
236 
237 
238 
239 
240 
241 
242 
243 
244 
245 
246 
247 
248 
249 
250 
251 
252 
253 
254 
255 
256 
257 
258 
259 
260 
261 
262 
263 
264 
265 
266 
267 
268 
269 
270 
271 
272 
273 
274 
275 
276 
277 
278 
279 
280 
281 
282 
283 
284 
285 
286 
287 
288 
289 
290 
291 
292 
293 
294 
295 
296 
297 
298 
299 
300 
301 
302 
303 
304 
305 
306 
307 
308 
309 
310 
311 
312 
313 
314 
315 
316 
317 
318 
319 
320 
321 
322 
323 
324 
325 
326 
327 
328 
329 
330 
331 
332 
333 
334 
335 
336 
337 
338 
339 
340 
341 
342 
343 
344 
345 
346 
347 
348 
349 
350 
351 
352 
353 
354 
355 
356 
357 
358 
359 
360 
361 
362 
363 
364 
365 
366 
367 
368 
369 
370 
371 
372 
373 
374 
375 
376 
377 
378 
379 
380 
381 
382 
383 
384 
385 
386 
387 
388 
389 
390 
391 
392 
393 
394 
395 
396 
397 
398 
399 
400 
401 
402 
403 
404 
405 
406 
407 
408 
409 
410 
411 
412 
413 
414 
415 
416 
417 
418 
419 
420 
421 
422 
423 
424 
425 
426 
427 
428 
429 
430 
431 
432 
433 
434 
435 
436 
437 
438 
439 
440 
441 
442 
443 
444 
445 
446 
447 
448 
449 
450 
451 
452 
453 
454 
455 
456 
457 
458 
459 
460 
461 
462 
463 
464 
465 
466 
467 
468 
469 
470 
471 
472 
473 
474 
475 
476 
477 
478 
479 
480 
481 
482 
483 
484 
485 
486 
487 
488 
489 
490 
491 
492 
493 
494 
495 
496 
497 
498 
499 
500 
501 
502 
503 
504 
505 
506 
507 
508 
509 
510 
511 
512 
513 
514 
515 
516 
517 
518 
519 
520 
521 
522 
523 
524 
525 
526 
527 
528 
529 
530 
531 
532 
533 
534 
535 
536 
537 
538 
539 
540 
541 
542 
543 
544 
545 
546 
547 
548 
549 
550 
551 
552 
553 
554 
555 
556 
557 
558 
559 
560 
561 
562 
563 
564 
565 
566 
567 
568 
569 
570 
571 
572 
573 
574 
575 
576 
577 
578 
579 
580 
581 
582 
583 
584 
585 
586 
587 
588 
589 
589 
590 
591 
592 
593 
594 
595 
596 
597 
598 
599 
600 
601 
602 
603 
604 
605 
606 
607 
608 
609 
610 
611 
612 
613 
614 
615 
616 
617 
618 
619 
620 
621 
622 
623 
624 
625 
626 
627 
628 
629 
630 
631 
632 
633 
634 
635 
636 
637 
638 
639 
639 
640 
641 
642 
643 
644 
645 
646 
647 
648 
649 
649 
650 
651 
652 
653 
654 
655 
656 
657 
658 
659 
659 
660 
661 
662 
663 
664 
665 
666 
667 
668 
669 
669 
670 
671 
672 
673 
674 
675 
676 
677 
678 
679 
679 
680 
681 
682 
683 
684 
685 
686 
687 
688 
689 
689 
690 
691 
692 
693 
694 
695 
696 
697 
698 
699 
699 
700 
701 
702 
703 
704 
705 
706 
707 
708 
709 
709 
710 
711 
712 
713 
714 
715 
716 
717 
718 
719 
719 
720 
721 
722 
723 
724 
725 
726 
727 
728 
729 
729 
730 
731 
732 
733 
734 
735 
736 
737 
738 
739 
739 
740 
741 
742 
743 
744 
745 
746 
747 
748 
749 
749 
750 
751 
752 
753 
754 
755 
756 
757 
758 
759 
759 
760 
761 
762 
763 
764 
765 
766 
767 
768 
769 
769 
770 
771 
772 
773 
774 
775 
776 
777 
778 
779 
779 
780 
781 
782 
783 
784 
785 
786 
787 
788 
789 
789 
790 
791 
792 
793 
794 
795 
796 
797 
798 
799 
799 
800 
801 
802 
803 
804 
805 
806 
807 
808 
809 
809 
810 
811 
812 
813 
814 
815 
816 
817 
818 
819 
819 
820 
821 
822 
823 
824 
825 
826 
827 
828 
829 
829 
830 
831 
832 
833 
834 
835 
836 
837 
838 
839 
839 
840 
841 
842 
843 
844 
845 
846 
847 
848 
849 
849 
850 
851 
852 
853 
854 
855 
856 
857 
858 
859 
859 
860 
861 
862 
863 
864 
865 
866 
867 
868 
869 
869 
870 
871 
872 
873 
874 
875 
876 
877 
878 
879 
879 
880 
881 
882 
883 
884 
885 
886 
887 
888 
889 
889 
890 
891 
892 
893 
894 
895 
896 
897 
898 
899 
899 
900 
901 
902 
903 
904 
905 
906 
907 
908 
909 
909 
910 
911 
912 
913 
914 
915 
916 
917 
918 
919 
919 
920 
921 
922 
923 
924 
925 
926 
927 
928 
929 
929 
930 
931 
932 
933 
934 
935 
936 
937 
938 
939 
939 
940 
941 
942 
943 
944 
945 
946 
947 
948 
949 
949 
950 
951 
952 
953 
954 
955 
956 
957 
958 
959 
959 
960 
961 
962 
963 
964 
965 
966 
967 
968 
969 
969 
970 
971 
972 
973 
974 
975 
976 
977 
978 
979 
979 
980 
981 
982 
983 
984 
985 
986 
987 
988 
989 
989 
990 
991 
992 
993 
994 
995 
996 
997 
998 
999 
999 
1000 
1001 
1002 
1003 
1004 
1005 
1006 
1007 
1008 
1009 
1009 
1010 
1011 
1012 
1013 
1014 
1015 
1016 
1017 
1018 
1019 
1019 
1020 
1021 
1022 
1023 
1024 
1025 
1026 
1027 
1028 
1029 
1029 
1030 
1031 
1032 
1033 
1034 
1035 
1036 
1037 
1038 
1039 
1039 
1040 
1041 
1042 
1043 
1044 
1045 
1046 
1047 
1048 
1049 
1049 
1050 
1051 
1052 
1053 
1054 
1055 
1056 
1057 
1058 
1059 
1059 
1060 
1061 
1062 
1063 
1064 
1065 
1066 
1067 
1068 
1069 
1069 
1070 
1071 
1072 
1073 
1074 
1075 
1076 
1077 
1078 
1079 
1079 
1080 
1081 
1082 
1083 
1084 
1085 
1086 
1087 
1088 
1089 
1089 
1090 
1091 
1092 
1093 
1094 
1095 
1096 
1097 
1098 
1098 
1099 
1100 
1101 
1102 
1103 
1104 
1105 
1106 
1107 
1108 
1109 
1109 
1110 
1111 
1112 
1113 
1114 
1115 
1116 
1117 
1118 
1119 
1119 
1120 
1121 
1122 
1123 
1124 
1125 
1126 
1127 
1128 
1129 
1129 
1130 
1131 
1132 
1133 
1134 
1135 
1136 
1137 
1138 
1139 
1139 
1140 
1141 
1142 
1143 
1144 
1145 
1146 
1147 
1148 
1149 
1149 
1150 
1151 
1152 
1153 
1154 
1155 
1156 
1157 
1158 
1159 
1159 
1160 
1161 
1162 
1163 
1164 
1165 
1166 
1167 
1168 
1169 
1169 
1170 
1171 
1172 
1173 
1174 
1175 
1176 
1177 
1178 
1179 
1179 
1180 
1181 
1182 
1183 
1184 
1185 
1186 
1187 
1188 
1188 
1189 
1190 
1191 
1192 
1193 
1194 
1195 
1196 
1197 
1197 
1198 
1199 
1199 
1200 
1201 
1202 
1203 
1204 
1205 
1206 
1207 
1208 
1209 
1209 
1210 
1211 
1212 
1213 
1214 
1215 
1216 
1217 
1218 
1219 
1219 
1220 
1221 
1222 
1223 
1224 
1225 
1226 
1227 
1228 
1229 
1229 
1230 
1231 
1232 
1233 
1234 
1235 
1236 
1237 
1238 
1239 
1239 
1240 
1241 
1242 
1243 
1244 
1245 
1246 
1247 
1248 
1249 
1249 
1250 
1251 
1252 
1253 
1254 
1255 
1256 
1257 
1258 
1259 
1259 
1260 
1261 
1262 
1263 
1264 
1265 
1266 
1267 
1268 
1269 
1269 
1270 
1271 
1272 
1273 
1274 
1275 
1276 
1277 
1278 
1279 
1279 
1280 
1281 
1282 
1283 
1284 
1285 
1286 
1287 
1288 
1289 
1289 
1290 
1291 
1292 
1293 
1294 
1295 
1296 
1297 
1297 
1298 
1299 
1299 
1300 
1301 
1302 
1303 
1304 
1305 
1306 
1307 
1308 
1309 
1309 
1310 
1311 
1312 
1313 
1314 
1315 
1316 
1317 
1318 
1319 
1319 
1320 
1321 
1322 
1323 
1324 
1325 
1326 
1327 
1328 
1329 
1329 
1330 
1331 
1332 
1333 
1334 
1335 
1336 
1337 
1338 
1339 
1339 
1340 
1341 
1342 
1343 
1344 
1345 
1346 
1347 
1348 
1349 
1349 
1350 
1351 
1352 
1353 
1354 
1355 
1356 
1357 
1358 
1359 
1359 
1360 
1361 
1362 
1363 
1364 
1365 
1366 
1367 
1368 
1369 
1369 
1370 
1371 
1372 
1373 
1374 
1375 
1376 
1377 
1378 
1379 
1379 
1380 
1381 
1382 
1383 
1384 
1385 
1386 
1387 
1388 
1389 
1389 
1390 
1391 
1392 
1393 
1394 
1395 
1396 
1397 
1397 
1398 
1399 
1399 
1400 
1401 
1402 
1403 
1404 
1405 
1406 
1407 
1408 
1409 
1409 
1410 
1411 
1412 
1413 
1414 
1415 
1416 
1417 
1418 
1419 
1419 
1420 
1421 
1422 
1423 
1424 
1425 
1426 
1427 
1428 
1429 
1429 
1430 
1431 
1432 
1433 
1434 
1435 
1436 
1437 
1438 
1439 
1439 
1440 
1441 
1442 
1443 
1444 
1445 
1446 
1447 
1448 
1449 
1449 
1450 
1451 
1452 
1453 
1454 
1455 
1456 
1457 
1458 
1459 
1459 
1460 
1461 
1462 
1463 
1464 
1465 
1466 
1467 
1468 
1469 
1469 
1470 
1471 
1472 
1473 
1474 
1475 
1476 
1477 
1478 
1479 
1479 
1480 
1481 
1482 
1483 
1484 
1485 
1486 
1487 
1488 
1489 
1489 
1490 
1491 
1492 
1493 
1494 
1495 
1496 
1497 
1497 
1498 
1499 
1499 
1500 
1501 
1502 
1503 
1504 
1505 
1506 
1507 
1508 
1509 
1509 
1510 
1511 
1512 
1513 
1514 
1515 
1516 
1517 
1518 
1519 
1519 
1520 
1521 
1522 
1523 
1524 
1525 
1526 
1527 
1528 
1529 
1529 
1530 
1531 
1532 
1533 
1534 
1535 
1536 
1537 
1538 
1539 
1539 
1540 
1541 
1542 
1543 
1544 
1545 
1546 
1547 
1548 
1549 
1549 
1550 
1551 
1552 
1553 
1554 
1555 
1556 
1557 
1558 
1559 
1559 
1560 
1561 
1562 
1563 
1564 
1565 
1566 
1567 
1568 
1569 
1569 
1570 
1571 
1572 
1573 
1574 
1575 
1576 
1577 
1578 
1579 
1579 
1580 
1581 
1582 
1583 
1584 
1585 
1586 
1587 
1588 
1589 
1589 
1590 
1591 
1592 
1593 
1594 
1595 
1596 
1597 
1597 
1598 
1599 
1599 
1600 
1601 
1602 
1603 
1604 
1605 
1606 
1607 
1608 
1609 
1609 
1610 
1611 
1612 
1613 
1614 
1615 
1616 
1617 
1618 
1619 
1619 
1620 
1621 
1622 
1623 
1624 
1625 
1626 
1627 
1628 
1629 
1629 
1630 
1631 
1632 
1633 
1634 
1635 
1636 
1637 
1638 
1639 
1639 
1640 
1641 
1642 
1643 
1644 
1645 
1646 
1647 
1648 
1649 
1649 
1650 
1651 
1652 
1653 
1654 
1655 
1656 
1657 
1658 
1659 
1659 
1660 
1661 
1662 
1663 
1664 
1665 
1666 
1667 
1668 
1669 
1669 
1670 
1671 
1672 
1673 
1674 
1675 
1676 
1677 
1678 
1679 
1679 
1680 
1681 
1682 
1683 
1684 
1685 
1686 
1687 
1688 
1689 
1689 
1690 
1691 
1692 
1693 
1694 
1695 
1696 
1697 
1697 
1698 
1699 
1699 
1700 
1701 
1702 
1703 
1704 
1705 
1706 
1707 
1708 
1709 
1709 
1710 
1711 
1712 
1713 
1714 
1715 
1716 
1717 
1718 
1719 
1719 
1720 
1721 
1722 
1723 
1724 
1725 
1726 
1727 
1728 
1729 
1729 
1730 
1731 
1732 
1733 
1734 
1735 
1736 
1737 
1738 
1739 
1739 
1740 
1741 
1742 
1743 
1744 
1745 
1746 
1747 
1748 
1749 
1749 
1750 
1751 
1752 
1753 
1754 
1755 
1756 
1757 
1758 
1759 
1759 
1760 
1761 
1762 
1763 
1764 
1765 
1766 
1767 
1768 
1769 
1769 
1770 
1771 
1772 
1773 
1774 
1775 
1776 
1777 
1778 
1779 
1779 
1780 
1781 
1782 
1783 
1784 
1785 
1786 
1787 
1788 
1789 
1789 
1790 
1791 
1792 
1793 
1794 
1795 
1796 
1797 
1797 
1798 
1799 
1799 
1800 
1801 
1802 
1803 
1804 
1805 
1806 
1807 
1808 
1809 
1809 
1810 
1811 
1812 
1813 
1814 
1815 
1816 
1817 
1818 
1819 
1819 
1820 
1821 
1822 
1823 
1824 
1825 
1826 
1827 
1828 
1829 
1829 
1830 
1831 
1832 
1833 
1834 
1835 
1836 
1837 
1838 
1839 
1839 
1840 
1841 
1842 
1843 
1844 
1845 
1846 
1847 
1848 
1849 
1849 
1850 
1851 
1852 
1853 
1854 
1855 
1856 
1857 
1858 
1859 
1859 
1860 
1861 
1862 
1863 
1864 
1865 
1866 
1867 
1868 
1869 
1869 
1870 
1871 
1872 
1873 
1874 
1875 
1876 
1877 
1878 
1879 
1879 
1880 
1881 
1882 
1883 
1884 
1885 
1886 
1887 
1888 
1889 
1889 
1890 
1891 
1892 
1893 
1894 
1895 
1896 
1897 
1897 
1898 
1899 
1899 
1900 
1901 
1902 
1903 
1904 
1905 
1906 
1907 
1908 
1909 
1909 
1910 
1911 
1912 
1913 
1914 
1915 
1916 
1917 
1918 
1919 
1919 
1920 
1921 
1922 
1923 
1924 
1925 
1926 
1927 
1928 
1929 
1929 
1930 
1931 
1932 
1933 
1934 
1935 
1936 
1937 
1938 
1939 
1939 
1940 
1941 
1942 
1943 
1944 
1945 
1946 
1947 
1948 
1949 
1949 
1950 
1951 
1952 
1953 
1954 
1955 
1956 
1957 
1958 
1959 
1959 
1960 
1961 
1962 
1963 
1964 
1965 
1966 
1967 
1968 
1969 
1969 
1970 
1971 
1972 
1973 
1974 
1975 
1976 
1977 
1978 
1979 
1979 
1980 
1981 
1982 
1983 
1984 
1985 
1986 
1987 
1988 
1989 
1989 
1990 
1991 
1992 
1993 
1994 
1995 
1996 
1997 
1998 
1999 
1999 
2000 
2001 
2002 
2003 
2004 
2005 
2006 
2007 
2008 
2009 
2009 
2010 
2011 
2012 
2013 
2014 
2015 
2016 
2017 
2018 
2019 
2019 
2020 
2021 
2022 
2023 
2024 
2025 
2026 
2027 
2028 
2029 
2029 
2030 
2031 
2032 
2033 
2034 
2035 
2036 
2037 
2038 
2039 
2039 
2040 
2041 
2042 
2043 
2044 
2045 
2046 
2047 
2048 
2049 
2049 
2050 
2051 
2052 
2053 
2054 
2055 
2056 
2057 
2058 
2059 
2059 
2060 
2061 
2062 
2063 
2064 
2065 
2066 
2067 
2068 
2069 
2069 
2070 
2071 
2072 
2073 
2074 
2075 
2076 
2077 
2078 
2079 
2079 
2080 
2081 
2082 
2083 
2084 
2085 
2086 
2087 
2088 
2089 
2089 
2090 
2091 
2092 
2093 
2094 
2095 
2096 
2097 
2097 
2098 
2099 
2099 
2100 
2101 
2102 
2103 
2104 
2105 
2106 
2107 
2108 
2109 
2109 
2110 
2111 
2112 
2113 
2114 
2115 
2116 
2117 
2118 
2119 
2119 
2120 
2121 
2122 
2123 
2124 
2125 
2126 
2127 
2128 
2129 
2129 
2130 
2131 
2132 
2133 
2134 
2135 
2136 
2137 
2138 
2139 
2139 
2140 
2141 
2142 
2143 
2144 
2145 
2146 
2147 
2148 
2149 
2149 
2150 
2151 
2152 
2153 
2154 
2155 
2156 
2157 
2158 
2159 
2159 
2160 
2161 
2162 
2163 
2164 
2165 
2166 
2167 
2168 
2169 
2169 
2170 
2171 
2172 
2173 
2174 
2175 
2176 
2177 
2178 
2179 
2179 
2180 
2181 
2182 
2183 
2184 
2185 
2186 
2187 
2188 
2189 
2189 
2190 
2191 
2192 
2193 
2194 
2195 
2196 
2197 
2197 
2198 
2199 
2199 
2200 
2201 
2202 
2203 
2204 
2205 
2206 
2207 
2208 
2209 
2209 
2210 
2211 
2212 
2213 
2214 
2215 
2216 
2217 
2218 
2219 
2219 
2220 
2221 
2222 
2223 
2224 
2225 
2226 
2227 
2228 
2229 
2229 
2230 
2231 
2232 
2233 
2234 
2235 
2236 
2237 
2238 
2239 
2239 
2240 
2241 
2242 
2243 
2244 
2245 
2246 
2247 
2248 
2249 
2249 
2250 
2251 
2252 
2253 
2254 
2255 
2256 
2257 
2258 
2259 
2259 
2260 
2261 
2262 
2263 
2264 
2265 
2266 
2267 
2268 
2269 
2269 
2270 
2271 
2272 
2273 
2274 
2275 
2276 
2277 
2278 
2279 
2279 
2280 
2281 
2282 
2283 
2284 
2285 
2286 
2287 
2288 
2289 
2289 
2290 
2291 
2292 
2293 
2294 
2295 
2296 
2297 
2297 
2298 
2299 
2299 
2300 
2301 
2302 
2303 
2304 
2305 
2306 
2307 
2308 
2309 
2309 
2310 
2311 
2312 
2313 
2314 
2315 
2316 
2317 
2318 
2319 
2319 
2320 
2321 
2322 
2323 
2324 
2325 
2326 
2327 
2328 
2329 
2329 
2330 
2331 
2332 
2333 
2334 
2335 
2336 
2337 
2338 
2339 
2339 
2340 
2341 
2342 
2343 
2344 
2345 
2346 
2347 
2348 
2349 
2349 
2350 
2351 
2352 
2353 
2354 
2355 
2356 
2357 
2358 
2359 
2359 
2360 
2361 
2362 
2363 
2364 
2365 
2366 
2367 
2368 
2369 
2369 
2370 
2371 
2372 
2373 
2374 
2375 
2376 
2377 
2378 
2379 
2379 
2380 
2381 
2382 
2383 
2384 
2385 
2386 
2387 
2388 
2389 
2389 
2390 
2391 
2392 
2393 
2394 
2395 
2396 
2397 
2397 
2398 
2399 
2399 
2400 
2401 
2402 
2403 
2404 
2405 
2406 
2407 
2408 
2409 
2409 
2410 
2411 
2412 
2413 
2414 
2415 
2416 
2417 
2418 
2419
```





Challenges

- ERD creation
- Create table scripts
- Team roles and balance of responsibility
- Avoiding metadata issues

Lessons learned

- Deep dive into rDBMS creation
- Workflow with a team, roles of each team member
- Specificity and quality of output on a busy work schedule
- Fully Understanding the task at hand before diving into it

Individual work

Collaboration

Final output

Thanks Everyone!

Any Questions?

