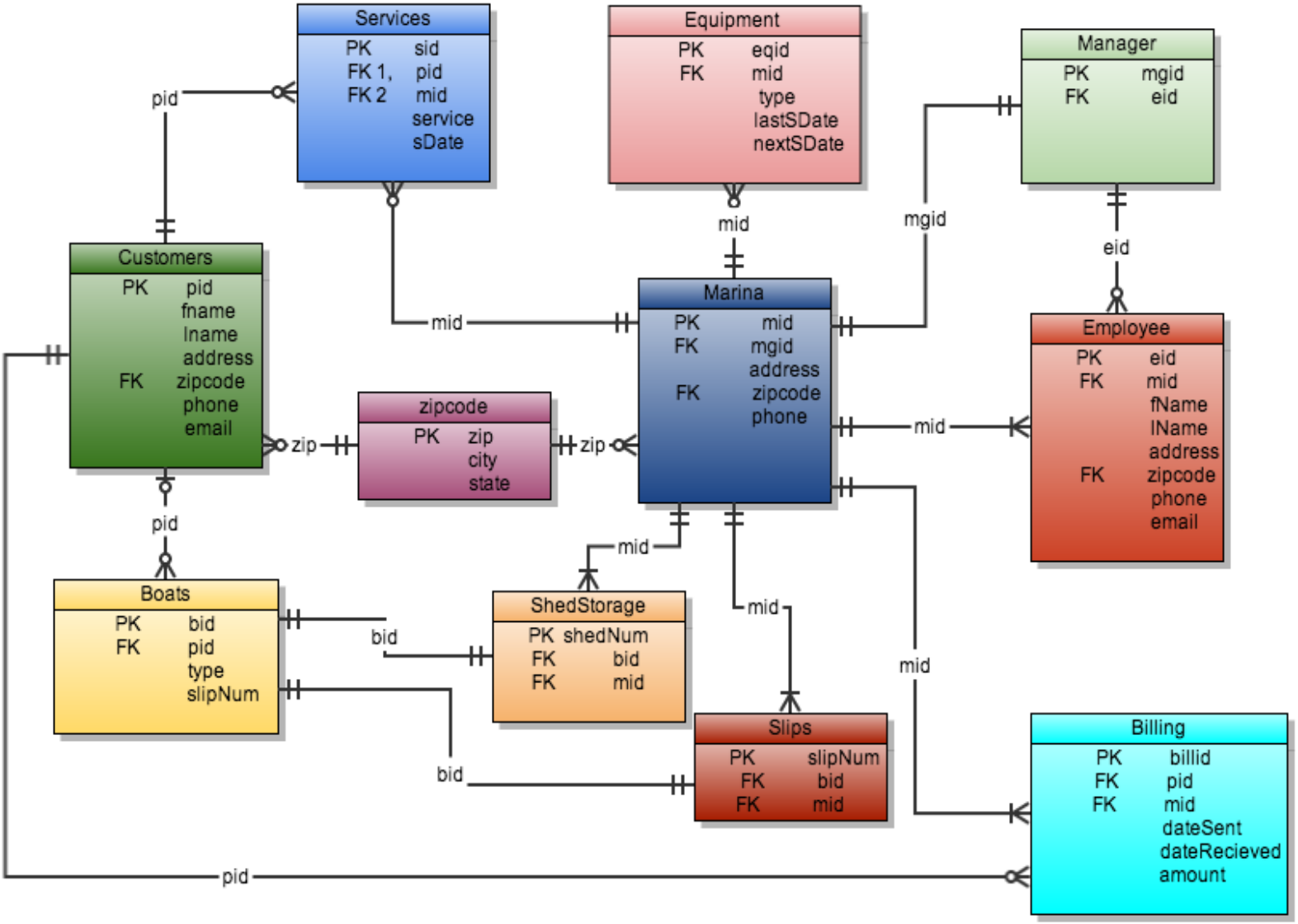
Brewer Yacht Yards & Marinas Database

Designed By Skylar Senning

Executive Summary

The Brewer Yacht Yards & Marinas Database is meant to streamline and organize day to day operations. These marinas fill the Connecticut coast line and deal with thousands of customers, boats, and services each year. This database makes marina operations less complicated by keeping track of boats and equipment so that yard managers know exactly where a customer's boat or a certain piece of machinery is at any point in the day. Every customer or boat owner is assigned an ID, which is then carried over to any service they might want done to their boat and also the bills which are sent out.

When spring hits every customer wants their boat in the water as fast as possible. The objective of this database make the lives of employees easier by managing bills, equipment service records, marina services, customer records, employee records, and boat locations all in one database rather then in file cabinets or digital marina maps.


Services
PK sid
FK 1, pid
FK 2 mid
service
sDate
Equipment
PK eqid
FK mid
type
lastSDate
nextSDate
Manager
PK mgid
FK eid
Customers
PK pid
fname
lname
address
FK zipcode
phone
email
zipcode
PK zip
city
state
Marina
PK mid
FK mgid
address
FK zipcode
phone
Employee
PK eid
FK mid
fName
lName
address
FK zipcode
phone
email
Boats
PK bid
FK pid
type
slipNum
ShedStorage
PK shedNum
FK bid
FK mid
Slips
PK slipNum
FK bid
FK mid
Billing
PK billid
FK pid
FK mid
dateSent
dateRecieved
amount
pid
mid
mgid
eid
zip
bid

ER Diagram

Customers Table

This table holds all the customers that use the services that the Marinas and Yacht Yards offer. Every customer is assigned a Person ID (pid) and is the primary key for the table. Other identifying information is also included like first name, last name, and address.

```
DROP table if exists customers;
CREATE TABLE customers (
            pid       char(4) not null,
            fName     char(20) not null,
            lName     char(20) not null,
            address   varchar(40) not null,
            zipcode   varchar(10) not null references zipcode(zip),
            phone       varchar(20) not null,
            email                   varchar(50),
primary key(pid)
);
```

Functional Dependencies

Pid → fname, lname, address, zipcode, phone email

pid character(4)	fname character(20)	lname character(20)	address character varying(40)	zipcode character varying(10)	phone character varying(20)	email character varying(50)
1	Skylar	Senning	41 River Road	06426	8603042624	skylarsenning@yah
2	Peter	Dixon	51 Pond Road	06417	8605752013	peterdixon@yahoo.
3	Clay	Arnold	5 Maint St	06475	8607675015	clayarnold@aol.co
4	Ian	Lemely	14 Hill Road	06426	8605162013	ianlemely@comcast
5	Brain	Shepherd	59 Medow Road	06475	8602034189	brianshepard@yaho
6	Maggie	McCarthy	16 Ceder Road	06417	8602034459	MaggieMcCarthy@me
7	Hunter	Lexa	2 West Main St	06426	8603045779	Hunter.Lexa124@yal

Marina Table

This table holds all the Brewers Marinas/Yacht Yards. Each location is assigned a marina ID (mid). For every marina there is a yard manager who is in charge of day to day operations

```
DROP TABLE IF EXISTS marina;
CREATE TABLE marina (
            mid       char(4) not null,
            mgid      char(4) not null references manager(mgid),
            address   varchar(40),
            zipcode   varchar(10) not null references zipcode(zip),
            phone         integer not null,
primary key(mid)
);
```

Functional Dependencies

mid →address, zipcode, phone

mid character(4)	mgid character(4)	address character varying(40)	zipcode character	phone integer
M001	MG01	15 North Main St	06426	7678267
M002	MG02	22 Essex Road	06475	3883260

Employee Table

This table holds all the employees at any of the marinas. Each employee is assigned a employee id (eid). Marina id is tied to the employee record so that managers know who works where incase they need more help or specialized skills

Functional Dependencies

eid → fname, lname, address, zipcode, phone, email

```
DROP TABLE IF EXISTS employee;
CREATE TABLE employee (
            eid         char(4) not null,
            mid         char(4) not null references marina(mid),
            fName       char(20) not null,
            lName       char(20) not null,
            address     varchar(50) not null,
            zipcode     varchar(10) not null references zipcode(zip),
            phone       varchar(20),
            email       varchar(50),
primary key(eid,mid)
);
```

eid character(4)	mid character(4)	fname character(20)	lname character(20)	address character varying(50)	zipcode character v	phone character varyi	email character varying(50)
1	M001	Ufuk	Flood	3175 Harvest Glade	06417	8605752013	rin.stone@geocities.com
2	M001	Euphemia	Luper	1538 Honey Zephyr Mountain	06426	4759063147	thor.streets@msn.com
3	M001	Sampson	Rine	8200 Hidden Spring Ledge	06426	2039210286	het-brus@webmine.com
4	M001	Hanley	Parshall	1395 Burning Park	06417	8603957875	vo-rem@carmag.com
5	M002	Amadeus	Dixon	6564 Little Brook Trace	06475	2035841358	eto_mees@mail.com
6	M002	Bentz	Talos	2004 Dusty Leaf Hollow	06475	2038346928	el-le@freespace.com
7	M002	Peter	Standard	923 Old Elk Meadow	06417	2035841358	nee-burn@freespace.com
8	M002	Alex	Favin	910 Thunder Mount	06426	8603957875	corin-tedes@myspace.com
9	M002	Chris	Huggert	2966 Heather Shadow Grounds	06475	2039210286	apa_garmon@infoseller.com

Manager Table

This table holds all the mangers for the marinas owned by Brewers. Every marina manager is assigned a managers id (mgid) along with their employee id.

```
DROP TABLE IF EXISTS manager;
CREATE TABLE manager (
            mgid        char(4) not null,
            eid         char(4) not null references employee(eid),
primary key(mgid)
);
```

mgid character(4)	eid character(4)
MG01	9
MG02	8

Services Table

This table holds all the service records for the marinas. Every service done is given a service id (sid) and is tied to the customer id (pid), as well as marina id (mid). The type of service is also listed along with the date that it was done.

Functional Dependencies

sid → service, date

sid character(4)	pid chara	mid character(	service character varying(60)	sdone date
S001	3	M001	Mast Removal	2013-09-10
S002	4	M001	Travel Lift Launch	2013-06-02
S003	7	M001	Engine Comissioning	2013-05-13
S004	1	M002	Engine Decomissioning	2013-10-15
S005	2	M002	Mast Rigging	2013-05-31
S006	6	M001	Hull Repair	2013-07-13
S007	1	M002	Bottom Painted	2013-04-20
S008	1	M002	Engine Decomissioning	2013-09-25

```
DROP TABLE IF EXISTS services;
CREATE TABLE services (
            sid         char(4) not null,
            pid         char(4) not null references customers(pid),
            mid         char(4) not null references marina(mid),
            service     varchar(60) not null,
            sDone       date not null,
primary key(sid)
);
```

Boats Table

This table holds all the boats that customers own, which are stored at marina location. Every boat is assigned a boat id (bid).

bid character(4)	pid character(4)	type character varying(40)	make character varying(40)	length integer
0001	2	Sail	John Alden	49
0002	7	Motor	Mako	29
0003	4	Sail	Viper	31
0004	5	Sail	Catalina	31
0005	7	Motor	Boston Whaler	19
0006	2	Sail	Benatue	75
0007	6	Sail	S & S	35
0008	5	Motor	Tarpon	54
0009	1	Motor	Nordic Tug	30
0010	2	Motor	John Alden	49
0011	3	Motor	Boston Whaler	30
0012	4	Sail	J 105	30
0013	7	Sail	J 105	30
0014	2	Sail	J 24	24
0015	1	Motor	Trophy	29
0016	4	Motor	Trophy	20
0017	3	Motor	Mako	31

Functional Dependencies

bid→ type, make, length

```
DROP TABLE IF EXISTS boats;
CREATE TABLE boats (
            bid         char(4) not null,
            pid         char(4) not null references customers(pid),
            type        varchar(40) not null,
            make        varchar(40) not null,
            lengthft      int not null,
primary key(bid)
);
```

Equipment Table

Managing and servicing equipment is crucial to the success of any yacht yard. This table holds all the equipment that the two marinas own, and assigns each piece of machinery an equipment id (eqid), which is tied to marina id so that employees and managers know where things are.

Functional Dependencies

eqid →type, lastDate, nextDate

```
DROP TABLE IF EXISTS equipment;
CREATE TABLE equipment (
            eqid        char(4) not null,
            mid         char(4) not null references marina(mid),
            type        varchar(50),
            lastSDate   date,
            nextSDate   date,
primary key(eqid)
);
```

eqid character(4)	mid character(4)	type character varying(50)	lastsdate date	nextsdate date
1	M001	Travel Lift CC203	2013-06-02	2014-04-06
2	M001	Liftall HTMS-180	2012-03-06	2014-03-06
3	M001	Taylor M-TSE-120-01	2013-05-13	2015-05-13
4	M001	YardArm Boat Jack	2012-05-23	2014-05-23
5	M002	Boat Hoist	2013-07-30	2015-07-30
6	M002	Hostar HHT 4200	2013-08-31	2015-08-31
7	M002	Brownell 60 Ton Semi	2012-08-15	2014-08-15
8	M002	Ascom BHT 40 ES	2013-05-19	2015-05-19

Billing Table

Organized billing is very important to the success of a marina because so many things are going on at once. Each bill is given a bill id (billid) and date sent/ received are recorded to keep track of payments

Functional Dependencies

billid →amount, datesent, daterecieved

```
DROP TABLE IF EXISTS billing;
CREATE TABLE billing (
            billid                  char(4) not null,
            pid                     char(4) not null references customers(pid),
            mid                     char(4) not null references marina(mid),
            amount                  numeric(10,2),
            dateSent                date not null,
            dateRecieved            date,
primary key(billid)
);
```

billid character(4)	pid character(4)	mid character(4)	amount numeric(10,2)	datesent date	daterecieved date
0001	2	M001	1159.95	2013-06-30	2013-07-06
0002	3	M001	5000.10	2013-06-30	2013-07-15
0003	6	M002	10000.00	2013-06-30	2013-11-21
0004	4	M001	100.95	2013-06-30	2013-07-01
0005	7	M002	2000.25	2013-06-30	2013-07-05
0006	1	M001	1159.95	2013-06-30	2013-07-06
0007	5	M001	10000.95	2013-06-30	2013-09-01

Shed Storage Table

During the winter many customers choose to store their boats in large sheds to protect the fiberglass from the elements. This table keep track of what boats are in what sheds at the two marinas

```
DROP TABLE IF EXISTS shedStorage;
CREATE TABLE shedStorage (
            shedNum     varchar(4) not null,
            bid         char(4) not null references boats(bid),
            mid         char(4) not null references marina(mid),
primary key(shedNum, mid)
);
```

shednum character varying(4)	bid character(4)	mid character(4)
SD01	0001	M001
SD02	0002	M001
SD03	0003	M001
SD05	0004	M001
SD03	0005	M002
SD01	0006	M002
SD02	0007	M002
SD06	0008	M002

Slip Table

A major portion of profits made by the marinas are from spring/summer slip rentals. Each slip is assigned a number that is tied to the boat id and marina id.

```
DROP TABLE IF EXISTS slips;
CREATE TABLE slips (
            slipNum     varchar(4) not null,
            bid         char(4) not null references boats(bid),
            mid         char(4) not null references marina(mid),
primary key(slipNum, mid)
);
```

slipnum character varying(4)	bid character(4)	mid character(4)
S001	0009	M001
S002	0010	M001
S005	0011	M001
S004	0012	M001
S001	0013	M002
S002	0014	M002
S003	0015	M002
S004	0016	M002
S005	0017	M002

Zip Code Table

This table holds the zip code, city and state for customers, employees and managers.

Functional Dependencies

zip → city, site

```
DROP table if exists zipcode;
CREATE TABLE zipcode (
            zip         varchar(10) not null,
            city        char(40) not null,
            state       char(40),
primary key(zip)
);
```

zip character varying(10)	city character(40)	state character(40)
06426	Essex	CT
06417	Deep River	CT
06475	Old Saybroc	CT

Views

This view shows the boats ordered by length with who owns them at Marina 2

```
CREATE VIEW sliplookup AS
SELECT distinct c.fName, c.lName, b.type, b.make, b.length, s.slipNum AS Slip
FROM  customers c, boats b, slips s
WHERE     c.pid = b.pid
  AND     b.bid = s.bid
  AND     mid = 'M002'
ORDER BY b.length
```

fname character(20)	lname character(20)	type character varying(40)	make character varying(40)	lengthft integer	slip character varying(4)
Ian	Lemely	Motor	Trophy	20	S004
Peter	Dixon	Sail	J 24	24	S002
Skylar	Senning	Motor	Trophy	29	S003
Hunter	Lexa	Sail	J 105	30	S001
Clay	Arnold	Motor	Mako	31	S005

Views

This view shows the service records done at Marina 1 ordered by date

```
CREATE VIEW servicelookup AS
SELECT distinct c.fName, c.lName, s.service, s.sDone, m.mid, m.address
FROM  customers c, services s, marina m
WHERE     c.pid = s.pid
   AND    s.mid = m.mid
   AND    s.mid = 'M001'
ORDER BY s.sDone;
```

fname character(20)	lname character(20)	service character varying(60)	sdone date	mid character(4)	address character varying(40)
Hunter	Lexa	Engine Comissioning	2013-05-13	M001	15 North Main St
Ian	Lemely	Travel Lift Launch	2013-06-02	M001	15 North Main St
Maggie	McCarthy	Hull Repair	2013-07-13	M001	15 North Main St
Clay	Arnold	Mast Removal	2013-09-10	M001	15 North Main St

Reports & Queries

```
SELECT c.fName, c.lName, bill.amount, bill.dateSent, bill.dateRecieved, bill.mid
FROM customers c, billing bill
WHERE c.pid = bill.pid AND bill.mid = 'M001'
ORDER by bill.dateRecieved;
```

This query shows the bills paid by customers at Marina 1 ordered by date received

fname character(20)	lname character(20)	amount numeric(10,2)	datesent date	daterecieved date	mid character(4)
Ian	Lemely	100.95	2013-06-30	2013-07-01	M001
Skylar	Senning	1159.95	2013-06-30	2013-07-06	M001
Peter	Dixon	1159.95	2013-06-30	2013-07-06	M001
Clay	Arnold	5000.10	2013-06-30	2013-07-15	M001
Brain	Shepherd	10000.95	2013-06-30	2013-09-01	M001

```
SELECT e.eqid, e.type, e.lastSdate, e.nextSdate, m.mid, m.address
FROM equipment e, marina m
WHERE      e.mid = m.mid
   AND     m.mid = 'M002'
ORDER by e.nextSdate;
```

This query shows the equipment and service records at Marina 2

eqid character(4)	type character varying(50)	lastsdate date	nextsdate date	mid character(4)	address character varying(40)
7	Brownell 60 Ton Semi	2012-08-15	2014-08-15	M002	22 Essex Road
8	Ascom BHT 40 ES	2013-05-19	2015-05-19	M002	22 Essex Road
5	Boat Hoist	2013-07-30	2015-07-30	M002	22 Essex Road
6	Hostar HHT 4200	2013-08-31	2015-08-31	M002	22 Essex Road

Stored Procedures

```
CREATE OR REPLACE FUNCTION billingamount() returns trigger
as $$
BEGIN
IF (billing.amount is null)
FROM billing
WHERE billing.amount is null
THEN
UPDATE billing SET amount = 0.00 WHERE amount IS NULL;
END if;
Return new;
END
$$LANGUAGE plpgsql;
```

This stored procedure updates the billing table if there is a NULL exists for any bill amount. It will update the amount to $0.00 so that management will clearly see that either the bill has been waived or a mistake was made during the the insert process

Triggers

```
Create Trigger NULLfix
After Insert or Update
On billing
FOR EACH ROW Execute
Procedure
billingamount();
```

Trigger is used to fix any NULLs that exist in the amount column with the billing table. When executed it runs the `billingamount();` procedure.

Security

This database will have three levels of security. First is the Amin(s) who have full access to the whole database, second is Managerial who have close to full access, and third is Employee who have limited access to the database.

Level 1 Access: Admin

```
Revoke all on Customers             from Admin;
Revoke all on Equipment             from Admin;
Revoke all on Employee              from Admin;
Revoke all on Marina                from Admin;
Revoke all on Boats                 from Admin;
Revoke all on Slips                 from Admin;
Revoke all on ShedStorage           from Admin;
Revoke all on Services              from Admin;
Revoke all on Zipcode               from Admin;
Revoke all on Manager               from Admin;
Revoke all on Billing               from Admin;

Grant insert, update, delete, select on Customers          to Admin;
Grant insert, update, delete, select on Equipment          to Admin;
Grant insert, update, delete, select on Employees          to Admin;
Grant insert, update, delete, select on Marina             to Admin;
Grant insert, update, delete, select on Boats              to Admin;
Grant insert, update, delete, select on ShedStorage        to Admin;
Grant insert, update, delete, select on Slips              to Admin;
Grant insert, update, delete, select on Zipcode            to Admin;
Grant insert, update, delete, select on Services           to Admin;
Grant insert, update, delete, select on Billing            to Admin;
Grant insert, update, delete, select on Manager            to Admin;
```

Security cont.

```
                        Level 2 Access: Managerial

Revoke all on Customers             from Manger008;
Revoke all on Equipment             from Manger008;
Revoke all on Employee              from Manger008;
Revoke all on Marina                from Manger008;
Revoke all on Boats                 from Manger008;
Revoke all on Slips                 from Manger008;
Revoke all on ShedStorage           from Manger008;
Revoke all on Services              from Manger008;
Revoke all on Zipcode               from Manger008;
Revoke all on Manager               from Manger008;
Revoke all on Billing               from Manger008;

Grant insert, update, delete, select on Customers          to Manager008;
Grant insert, update, delete, select on Equipment          to Manager008;
Grant insert, update, delete, select on Employees          to Manager008;
Grant select                          on Marina            to Manager008;
Grant insert, update, delete, select on Boats              to Manager008;
Grant insert, update, delete, select on ShedStorage        to Manager008;
Grant insert, update, delete, select on Slips              to Manager008;
Grant insert, update, delete, select on Zipcode            to Manager008;
Grant insert, update, delete, select on Services           to Manager008;
Grant insert, update, delete, select on Billing            to Manager008;
Grant select                          on Manager           to Manager008;
;
```

Security cont.

```
                        Level 3 Access: Employee

Revoke all on Customers             from Employee001;
Revoke all on Equipment             from Employee001;
Revoke all on Employee              from Employee001;
Revoke all on Marina                from Employee001;
Revoke all on Boats                 from Employee001;
Revoke all on Slips                 from Employee001;
Revoke all on ShedStorage           from Employee001;
Revoke all on Services              from Employee001;
Revoke all on Zipcode               from Employee001;
Revoke all on Manager               from Employee001;
Revoke all on Billing               from Employee001;

Grant insert, update, select          on Customers          to Employee001;
Grant insert, update, delete, select on Equipment           to Employee001;
Grant select                          on Employees          to Employee001;
Grant select                          on Marina             to Employee001;
Grant insert, update, delete, select on Boats               to Employee001;
Grant insert, update, delete, select on ShedStorage         to Employee001;
Grant insert, update, delete, select on Slips               to Employee001;
Grant insert, update, delete, select on Zipcode             to Employee001;
Grant insert, update, delete, select on Services            to Employee001;
Grant insert, update, select          on Billing            to Employee001;
```

Implementation Notes

- Implementation for this database should be simple
- Run the create statements and insert valid data
- Change any of the grant/revoke statements if necessary
- Change slip numbers/shed numbers to make moving boats easier
- Services, slip storage, shed storage do not connect to billing because boats are constantly moving.

Known Problems

- The bill amount has to be manually computed due to the varying service charges and slip/storage fees. It could be automated if a concrete price list was developed.
- The shed and slip numbering for each marina may not be the same and could cause problems.