http://notewiki.gargantubig.com/index.php?title=Summer\_08

Day one 7-14-08

Quiz every wednesday including the 16th sort of non cumulative

Fair amount of homework

oracle 10g programming or you could get oracle 9i - sunderaman

database management systems - ramakrishnan get a 1200 page book on db management date - newer edition 6 or above

Download "oracle 10g download" free until you plan to make some money

Generation languages

1GL machine language

2GL Assembler language

3GL Higher Level Languages

4GL DBMS Oracle DB2 SQLServer Mysql Postgres Derby...

**buzzwords**

**Integrity - data should be correct**

Authentication - to connect to the db

Authorization - to do stuff in the db

Constraints - sallary >= 0

**Persistence - making sure the data stays around**

backup and recovery RAC Standby

3 tier architecture --> 1 browser -2 middleware -3 db

**Independence**

separation between conceptual and internal

People

DA - Data Administrator - makes decisions over what things to keep in db, and who is authorized to do what in the db, what information goes in and out. -Policy -non technical

DBA - Database Administrator manages the actual database, implements what the DA designed. -top of the hierarchy -usually certified -implementation - technical

Application Programmer Write software so users can get data into and out of the db. -What does the data look like to the end user. - technical - we can go towards this job. (hibernate - java to oracle oop conversion)

User Maybe non programmer, dont know how to use the db, etc.

levels of a database

**internal**

how the data is really stored.

**conceptual**

(logical) design (DA) implementation (DBA/Application Programmer)

**external**

views How it is seen. i.e. Table may only show some of the fields, but not others. Or fields from more than one db.

Types of DBMS

**Hierarchical**

**Linked**

==== Relational ==== RDBMS were concerned with this one ==== Object ==== ODBMS

==== Object relational ==== ORDBMS

Oracle can function as last 3... RDBMS ODBMS ORDBMS

**Spatial**

Part of your data is locations. (spring 600 level course)

**Temporal**

salary known as a function of time. substantially time stamped data. Good for reporting. Like prices and history

temporal and spatial databases get very large. words with large databases

data warehousing - how its stored. data mining - how to get stuff out.

subdivisions of db programming

1 DDL Data definition language CREATE TABLE DROP TABLE ALTER TABLE

2 DML Data Manipulation Language SELECT FROM INSERT INTO (updates) UPDATE SET (updates) DELETE FROM (updates)

3 DCL Data Control Language GRANT REVOKE

SQL CONNECTING

sqlplus st\*\*/cs514@cfedb.sdsu.edu

set pages 100 desc empbb02

alter table empbb02 drop column wife;

delete from deptbb02 where deptno = 89 or deptno = 44 ;

desc empbb02

desc salsbb02

CREATE TABLE sals (grade NUMBER(2), role VARCHAR2(8),

losal NUMBER(8),hisal NUMBER(10));

INSERT INTO sals VALUES(1,'staff',5001,9000);

select \* from sals;

scripting

script lab1.1

%sqlplus st\*\*/cs514@cfedb.sdsu.edu //your masc acct number in \*\*

desc sals

--set rows 100--

CTRL d to stop transcribing

make transcript one not null dont want to see all insert into's

sqlplus st53/cs514@cfedb.sdsu.edu

CREATE TABLE eeegggg (a NUMBER, b NUMBER NOT NULL);

desc eeegggg

squirle mail rohan

one not null on ename in mbubba02

day2 080716

how to go to host

sql> host oz! sql> host ls

>ctrl-d to get back

-- how to document what you're doing. SPOOL showass2 //SPOOL OFF to end simultaneous peripheral operations off line

oracle interpreter comments -- create 4 tables DROP TABLE ... //get rid of prev table CREATE TABLE ... DROP TABLE CREATE...

-- quiz1 --

**Theoretical underpinnings of relational DBMS's**

why did relational win? good theory. date's book

set theoretic and functional - mathematic

**Relational Models**

Relational Algebra Relational Calculus logic is prepositional calculus nands and nors predicate calculus - adds variables (predicate = relational)

-- relational algebra --

**Relational Operators**

===== Datename ===== ===== Datesyntax ===== ===== Sunderaman =====

rename s rename city as c Ps(R)(rename the whole table)

(in table s rename column city to c)

union UNION U (union symbol)

intersection INTERSECT upside down U (intersection symbol)

difference MINUS -

project R {attr, comma list} pi sub a(R)

(cross, cartesian) [pi sub ename,sal (e mpb02)]

restrict(select) R where {boolean expression} Sigma sub E(R)

product TIMES X

(natural)join RJOINS |><|

divide A DIVIDEBY B PER C R/S r%s horizontal divide

T .=. JOIN S  ?? what ??

What is a relation?

A set of attributes (columns) where an attribute is a pair consisting of 2 things : (attribute name, attribute type)

relational algebra example

Find last name first name and salary for all employees who make 80000 or more

Sunderaman version

Pie *sub(pname ename sal)* Sigma *sub(sal /> 8000)* (mpbb02 |><| infobb02)

Datesyntax version

R{pname, ename, sal} R where sal >=80000 (mpbb02 RJOINS infobb02)

Union Compatibility

when can you union things?

same name, same type

same number of columns same type of data.

**7-product**

if you take the product of a relation with 5 attributes and relation with 10 attributes you must get a table with 15

what if i get product of 2 copies of same table empbb03.sal use dot notation. rename the whole table

empbb03 rename ename as enameb then do product

works on columns.

R has {a sub(1)...,a sub(k),a sub(k+1),...,a sub(n)}

S has {a sub(1)...,a sub(k),b sub(k+1),...,b sub(t)}

R JOIN S has {a sub(1),...a sub(k),a sub(k+1),...,a sub(n), b sub(k+1),...,b sub(t)}

**9 divide**

R/S R divide by S

R has {a sub(1)...,a sub(k),a sub(k+1),...,a sub(n)}

 S has {a sub(k+1),...,a sub(n)}

r s

joe shirt shirt

joe pants pants

mike pants

bill shirt

bill pants

bill shoes

r/s --> you take joe and bill

union intersection and minus independent? yes A-B = A Union notB --> there is no not in relational algebra

**Application Time**

doing stuff in oracle instead of theory

host go back to rohan unix, ctrl-d return.

edit afiedt.buf edits last messed up statement?

-- get me people select ename, sal, pos from salsbb02

when you want to rename the columns

select ename as "last name",sal as salary, pos as position from empbb02

select ename as "last name" <-- attribute alias, temporary, " " when you ned a space in it

select \* from empbb02 WHERE sal>80000;

((most of the time in oracle you use single quotes)) exception, column alias

select ename from empbb02 where ename like '%w%';

anyone with a w in name

select fname, ename from empbb02,infobb02 where empbb02.empno = infobb02.empno

select e.empno, fname, ename from empbb02 e,infobb02 i where e.empno = i.empno

-- btitle 'report done' -- bottom title set headsep ! -- header separator ttitle 'First Example ! CS 514' -- top title spool fred.dat set echo on set pause on set pages 30 -- this is a comment select \* from empbb02; select \* from salsbb02; -- set echo off spool off --

equijoin is sql equivalent of a natural join

select \* from empbb02 NATURAL JOIN

select \* from empbb02, infobb02; (does a product)

create tttttt as select \* from empbb02,infobb02

select \* from empbb02 union select \* from empbb02

select \* from empbb02 where sal <40000 union select \* from empbb02 where sal >110000;

intersect minus

relational calculus 08 07 21

Relational Algebra.

Relational Calculus (tuple) we have variables that range over tuples

eg: X sub(s) means X ranges over predicate (table) S

One pass Lle notation. {Xsub(1).asub(1), Xsub(n).asub(n):W(xsub(1),...,xsub(n))} // W <-- wff // Xsub(1) <-- var

^ and v or

{X.ename, X.bass empbb02(Xsub(1)) ^ Xsub(1).sal>10000} chose all players whose sallary is greater than 10000

definition of atomic wff (weff?) simplest kind of weff

1. S(X), S a relational...mbil? x a variable 2. S a big theta Ssub(2) b where Ssub(1), Ssub(2) are relations, theta is oneof {=, not=, <, <=, >, >=} Ssub(1).a theta Constant.

**deff of a wff is one of (Well Formed Formula)**

(1) atomic wff

(2) negation symbol W where W is a wff

(3) (Wsub(1), VWsub(2)) Where Wsub(1), Wsub(2) are wff's

(4) there exists x in W where W is a wff.

a free variable escapes a thereexists or a for every.

**oracle portion**

// using db dual if null participates in arithmetic answer is allways null

user catalog == cat table view sequence are all objects

purge recyclebin drop table nnn purge

SELECT fname || ' ' || ename AS name FROM empbb02 NATURAL JOIN 2 infobb02;

SELECT ename from empbb02 WHERE sal BETWEEN..........and..........

NOT BETWEEN.........AND........

SELECT \* FROM empbb02 WHERE sal IN(2000,10000,85000); //is an element of this set

SELECT \* FROM empbb02 where sal NOT IN (2000,10000,85000);

**Wildcards**

'%a%' contains letter a

'\_a\_\_' 4 letters 2nd is a

SELECT \* from \_\_\_\_ ORDER BY ename desc (descending)

SELECT LOWER('OrAcLE') from dual; / makes stuff lower case

SELECT UPPER

SELECT INITCAP('OrAcLE') from dual

SELECT CONCAT

SELECT SUBSTR('facetious' 2,4) from dual; //substring start at 2 get 4 letters (first is 1)

SELECT length('facetious') from dual

SELECT INSTR('abracadabra','r')from dual; // first occurrence of r in abracadabra. (index of string in string)

 SELECT INSTR('abracadabra','ra',2,3); start at 2rd letter find the 3rd occurrence

SELECT ename , LPAD(sal,10,'\*') from empbb02 //pads on the left with \* to make cell be 10 wide total.

RPAD // same but pad on right

SELECT LTRIM('aabbababcab*ba') from dual; //remove leading a's or b's*

end of characters

Numeric...

SELECT ROUND (135.6789, 2) from dual; // round to the 2nd decimal place (wo parameter rounds to int.

SELECT CEIL(112.3) //ceiling

SELECT FLOOR

TRUNC -- truncates

SELECT SYSDATE from dual -- current date in the standard format DD-MON-YY

math SELECT SIN COS TAN ACOS ASIN ATAN MOD POWER EXP LN

**08 07 23**

select ename,sal + NVL (incentives,0) // replace nul values with

date functions

select round sysdate from dual;

select trunc(sysdate) from dual

SELECT MONTHS\_BETWEEN ('1-jan-2000', SYSDATE) from dual

SELECT ADD\_MONTHS (SYSDATE,16) from dual

SELECT NEXT\_DAY (SYSDATE,'tuesday') from dual

select last\_day (sysdate) from dual;

select (sysdate + 4) from dual;

TO\_DATE TO\_CHAR TO\_NUMBER

SELECT TO\_CHAR (SYSDATE,'MM/DD/YYYY) FROM DUAL;

SELECT TO\_CHAR (SYSDATE,'MON DD,YYYY) FROM DUAL;

TO\_CHAR (SAL,'L9,999,999) from empbb02 --(L does local currency symbol)

TO\_CHAR (SAL,'L0,000,000) from empbb02 --(L does local currency symbol)

DECODE

set pages 50

SELECT e.ename,e.sal,s.grade FROM empbb02 e, salsbb02 s WHERE e.deptno <30 AND s.role = 'hitter' AND e.sal BETWEEN s.losal AND s.hisal;

SELECT e.ename || ' works for ' ||e2.ename AS "Who's the Boss" FROM empbb02 e,empbb02 e2 WHERE e.boss =e2.empno;

<http://www.databasteknik.se/webbkursen/relalg-lecture/index.html>

**Joins**

Outer Join (full Outer Join, left Outer Join, right Outer Join)

should get the stuff thats missing left will take the stuff from the left table thats not in the right. full will get from both.

AAAA: A B BBBB: B C

10 5 5 3

20 10 30 8

normal join gets just the row with the 5 in b left outer join picks up the guy on the left 5, 10 rigth outer join picks up on the right

**Group Functions**

AVG SUM MAX MIN COUNT -- group functions

can be applied to a column result is a number

SELECT COUNT(\*) FROM infobb02;

SELECT COUNT(incentives) FROM empbb02;

SELECT SUM(sal) from empbb02;

SELECT AVG(sal) from empbb02;

SELECT AVG(nvl(incentives,0)) from empbb02; average with nulls included as 0

SELECT AVG(incentives) from empbb02;

select deptno, sum(sal) from empbb02 GROUP BY deptno;

select deptno, sum(sal) from empbb02 GROUP BY deptno ORDER BY deptno;

select deptno, sum(sal), sum(incentives) from empbb02 GROUP BY deptno ORDER BY deptno;

anything in the group by should have a group function applied

where is forbidden after a groupby change to "having"

select deptno, sum(sal), sum(incentives) from empbb02 HAVING deptno < 50 GROUP BY deptno ORDER BY deptno ;

**subquery / subselect**

SELECT ename from empbb02 where sal > (select sal from empbb02 WHERE empno = 735);

subselect is the second statement

single row query

SELECT ename from empbb02 where sal IN (select sal from empbb02 WHERE empno = 735 or empno = 777);

multiple row query people who make either same as 735 or 777

SELECT ename from empbb02 where sal >ALL (select sal from empbb02 WHERE empno = 735 or empno = 777);

SELECT ename from empbb02 where sal >ANY (select sal from empbb02 WHERE empno = 735 or empno = 777);

ALL = for every ---> ANY = there exists

SELECT ename , sal FROM empbb02 WHERE (sal,deptno) IN (SELECT sal,deptno FROM empbb02 WHERE sal > 10000 AND deptno>15);

**Primary Key**

candidate key any row thats unique and will continue to be unique even if the table changes. a minimum set of columns which uniquely identifies a row for any valid values for the columns.

minimum means if you take any one of them away then it will no longer be unique.

primary key is a selected candidate key

primary implies not null and unique.

how to establish a primary key

ALTER TABLE empbb02 ADD CONSTRAINT empbb02\_empno\_pk PRIMARY KEY (empno)

// makes the primary key empno -- pk

salsbb02 grade and role is a decent candidate key

PRIMARY KEY (grade,role)

after lab we will have 5 more not null's

INSERT INTO empbb02 VALUES('735','shmoe',null,null,null,null,null,null);

Primary key - specially selected primary key. candidate key - set of columns with any valid input that uniquely identify a row.

7-28-08 class notes

**Lab2 info**

infobb02 with boss on empbb02 boss with employee number on 2

Create TABLE t7 (a number PRIMARY KEY, b NUMBER)

pro risk of omission. cost no chosen name for constraint, so cannot easilly drop the constraint.

Create TABLE t7 (a number, b NUMBER, PRIMARY KEY (a,b)); //make both of them primary keys.

**data types**

VARCHAR2(SIZE)

CHAR(SIZE)

NUMBER(P,S)

DATE

LONG //(UP TO 2 GB)

CLOB //CHARACTER LONG OBJECT (up to 4gb)

RAW //NONCHARACTER DATA

LONGRAW // MORE RAW BINARY DATA

BLOB //Binary large object (picture, audio file etc)

BFILE //LIKE A BLOB IS KEPT IN EXTERNAL STORAGE (video file large)

NCLOB // national character set -- clob, language not english

DROP TABLE t CASCADE CONSTRAINTS //if stuff is dependent, drop them too

ALTER TABLE t DROP COLUMN a; // ditch a column. was not supported pre oracle 8

ALTER TABLE t SET UNUSED a;

ALTER TABLE t DROP UNUSED COLUMNS

ALTER TABLE t TRUNCATE TABLE //bring table back to new no rows

ALTER TABLE t DELETE TABLE //

how to copy a table -- used in lab 3

COPY // syntax ugly so dont use

sqlldr // better - a utility

**CONSTRAINTS**

NOT NULL, PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK

naming convention: nn pk fk uk ck

ALTER TABLE b ADD CONSTRAINT b\_c\_fk foreign KEY (c) REFERENCE a(c2) // cant make an entry in b into c unless the table a already has that value in column c2.

cannot delete a parent if there is a child pointing to it with a foreign key reference. Unless you say ON DELETE CASCADE, which means when you kill the parents you kill the children.

ALTER TABLE empbb02 ADD CONSTRAINT empbb02\_sal\_incentives\_ck

sal+incentives <= 1million

**Check Constraint**

CHECK (sal+incentives <= 1000000000)

Select constraint\_name, constraint\_type, search\_condition from user\_constraints where table\_name = 'EMPBB02' or table name = 'INFOBB02';

CREATE TABLE empbb02bkup AS SELECT \* FROM empbb02; //creates a copy of the table

DROP VIEW empview16

CREATE VIEW empview16 as select \* from empbb02 WITH READ ONLY

update empview16 set sal = 41000 WHERE ename = 'musial'

DESC user\_views //shows active views

CREATE OR REPLACE VIEW

CREATE SYNONYM empxxx FOR empbb02

DROP SYNONYM empxxx

DESC USER\_OBJECTS // stores the stuff

DESC user\_synonyms

CREATE SEQUENCE deptbb02\_deptno INCREMENT BY 5 START WITH 10 MAXVALUE 5000 NOCACHE NOCYCLE;

cycle prevents you going around again.

INSERT INTO deptbb02 VALUES (deptbb02\_deptno.NEXTVAL 'bribes','food','santee','never give up')

Class notes 07-30-2008

Set time on // puts the time in the task bar

set timing on // tells you how long it took.

select \* from deptbb02;

CREATE INDEX empbb02\_ename\_ndx ON empbb02(ename);

desc user\_objects

select object\_name FROM user\_objects

variables: substitution runtime (&x)

select ename from empbb02 where empno = &x;

enter value for x:

if the thing is number dont need single quotes, characters youll want to use single quotes

select ename from empbb02 WHERE empno= '&x';

enter value for x:

pseudo app accept - have to put in a script file.

-- use this to demonst

ACCEPT num PROMPT 'Please enter the age of an uncle: '

ACCEPT age PROMPT 'Please enter the age of an uncle: '

ACCEPT iq PROMPT 'Please enter the age of an uncle: '

INSERT into tp VALUES &num,&age,&iq

**sqlldr SQL\*Loader**

can work with a .csv. trouble is if there are commas in the file. or .psv pipe seperated values

creating a csv

spool pipe

select age||','||name||','||iq from tp;

vi pipe.lst

Use for lab3 to copy to other account

you'll have to create the tables by hand but then use sql ldr to populate them

you may have a lot of trailing blanks if your last column is a varchar2.

you can use an editor to get rid of trailing blanks "trailing null character problem"

when making csv you can rearrange the columns

syntax fixes

add a column of zeroes which you dont read from csv

permissions dir(1),user(3),group(3),world(3)

-rwx------ or -rwxrwxrwx

you will make an executable file to have the sqlldr commands in it.

files with x permission

sqlldr eckberg/carl@mlidb control = loadsals.ctl

sqlldr eckberg/carl@mlidb control = loademp.ctl

sqlldr eckberg/carl@mlidb control = loadinfo.ctl

sqlldr eckberg/carl@mlidb control = loaddept.ctl

chmod u+x copytables // unix script file

need to see the tables are created correctly, and populated correctly.

**Embedded coding**

PL/SQL can put sql statements in a loop. (can take a certification exam)

JAVA JDBC embedding sql into java is innefficient

JSQL oracle product translated down into jdbc to check syntax. did not catch on well

**pl/sql organization**

Block

Anonymous Named

Functions Procedures

declarations //optional declaration part

begin

(executables go here, must have one)

end

you can have an exception section before end

spool plsql.dat set echo on can use variable or var

global variables or bind variables. also external

VARIABLE mo\_sal VARCHAR(10);

VAR mo NUMBER

VAR date2 DATE := '03-JAN-07';

DECLARE

v\_sal NUMBER(9,2):=&ann\_sal;

v\_date DATE := '03-JAN-07';

BEGIN

 :mo\_sal:=TO\_CHAR(v\_sal/12,'99999.99');

 :mo := v\_sal  ;

END;

/

PRINT mo\_sal

set serveroutput on --// on test

DECLARE

g\_monthly\_sal NUMBER

v\_sal NUMBER(9,2) NOT NULL:+ &p\_annual\_sal;

c\_val CONSTANT NUMBER:= -10;

BEGIN

g\_monthly\_sal := v\_sal/12;

DBMS\_OUTPUT.PUT\_LINE('the monthly sallary is $'||

TO\_CHAR(g\_monthly\_sal)|| ' and your value to us is '

||c\_val||'.'); --// on test

END;

/

try slashes or dots or carriage returns

hope for: "pl/sql procedure successfully completed"

set pause on

set echo on

set pages 100

set serveroutput on

DECLARE

empname empbb02.ename%TYPE;

empmgr empbb02.boss%TYPE;

empsal empbb02.sal%TYPE;

BEGIN

SELECT emp.ename,emp.boss,emp.sal

INTO empname,empmgr,empsal

FROM empbb02 emp

WHERE emp.empno = 735

DBMS\_OUTPUT.PUT\_LINE('employee 735 is '||empname||', works for '||empmgr||

' and makes '||empsal);

END;

/

cant do a select without an into select into can only bring back a single row requires the WHERE

080804 class

DBMS.OUTPUT.PUT\_LINE(to\_char)(i)\\'. '\\spChow(i)); exit when spchow.last = i

CREATE OR REPLACE FUNCTION getDnamebb02(name IN empbb02.ename%TYPE)

RETURN deptbb02.dname%TYPE AS gfname deptbb02.dname%TYPE

BEGIN

SELECT dname

INTO gdname

FROM empbb02,deptbb02

WHERE empbb02.deptno = deptbb02.deptno AND name = ename;

RETURN (gdname);

END;

/

If theres an error it just says compiled with errors, adding"show errors" will give more detail on compile

show user procedures

desc functionname gives you lots of info on the function

DBMS OUTPUT PUT LINE -- know for wednesday

[Using SQL Loader](http://notewiki.gargantubig.com/index.php?title=Using_SQL_Loader)

Anonymous blocks

commit;

insert into\_deptbb02 values(70, 'snacks', null, null, 'get over it');

savepoint spot;

insert into deptbb02 values(90, 'bookclub', null, null, 'read it and weep');

rollback to spot;

//It was for setting a point restoring to that previous point

BEGIN

SELECT dname

INTO gdname

FROM empbb02, deptbb02

WHERE empbb02.deptno=deptbb02.deptno AND name = ename;

RETURN(gdname);

END

select \* from user\_procedures;

function / procedures that have been stored

-- this code invokes a stored function column “department name” format A15; spool getDnamebb02Test.dat set echo on set pages 1000000000SELECT ename,getDnamebb02(ename) AS “department name” FROM empbb02; set echo off quit

-- simple test of the 'execute' command, which basically creates a -- minimal anonymous block containing the following invocation VARIABLE deptname VARCHAR2(8); execute:deptname:=getDnamebb02('lasorda'); PRINT deptname

CREATE OR REPLACE FUNCTION getDnamebb02b(no IN NUMBER RETURN NUMBER AS avg1 NUMBER; tot NUMBER;

BEGIN

RETURN (avg1);

END;

/

SHOW ERRORS

CREATE OR REPLACE PROCEDURE empStatsbb02(

deptid IN empbb02.deptno%TYPE,

avgsal OUT NUMBER,

totalsal OUT NUMBER) AS

BEGIN

SELECT ROUND(AVG(empbb02.sal)),SUM(empbb02.sal)

INTO avgsal, totsal

FROM empbb02

WHERE empbb02.deptno=deptid;

END;

.

/

SHOW ERRORS

start empStatsbb02Test

set pages 100

DECLARE

avg1 NUMBER;

tot NUMBER;

BEGIN

empStatsbb02(20,avg1,tot);

DBMS\_OUTPUT.PUT\_LINE('for dept 20 the average salary is '||' and the total salary is '||tot);

END;

.

/

PRINT avg1

PRINT tot

[email sql8-4](http://notewiki.gargantubig.com/index.php?title=Email_sql8-4)

Class 080806

no top 5 on quizzes

exceptions will be tested

Class 080811

"jdbc:oracle:thin:cslabdb:1525:cfedb"

server port (dont need .sdsu.edu)

cslabdb - DNS name of his server thin client - not a lot of client side processing.

host cslabdb.sdsu.edu has address 130.191.28.142

try {

catch (Exception e) {System.out.println("MR.UnitSitQueries.constructor.Exception: "+ e);

stmt = con.createStatement(); resultset rs= stmt.executeQuery(query);

SELECT

 Leave the classes12.zip zipped or jarred

ORACLE\_HOME /opt/oracle

have a recognition for oracle driver names ie. classes12.jar .zip classes12dms.jar ojdbc14.jar ojdbc14\_g.jar ojdbc14dms.jar ojdbc14dms\_g.jar nls\_charset12.jar

ODBC open database connectivity.

file: 7 jdbc7Ora.java connection conn try oracle thin driver

preparedStatement pstmt = conn.preparedStatement(INSERT INTO deptbb02 VALUES(?,?,?,?,?);

know for test: getstring getint setstring setint

jdbc7bOra.java -- nothing much

jdbc8Ora.java -- prepared statemnt- same as 6 structured programming loop boolean done = false while (!done) if deptno == 0 done = true

deptno= integer.parseInt(getData.readline());

conn.preparestatement conn.preparecall callableStatement

javac, then java

jdbc9Ora

 jdbc9bOra -- nothing interesting

Question: callable statement execute update call

jdbc10Ora --

resultset metadata object.

what tables does it have, what types of fields... info on the schema.

certain methods that can be sent to a database metadata.

 jdbc13Ora--

DatabaseMetaData dbmd = conn.getMetaData(); sopln. dbmd.getDatabaseProductname()); dbmd.getDriverName()

getTables(...) lots of parameters... need something to get usefull info

there is a publication about jdbc that talks about jdbc. 150pages of database metadata. need the book if you want to be a good user of database metadata.

jdbc14Ora -- dbmd.getSchemas()

default test user: Scott pw:Tiger on every oracle db

getProcedures() --

procedures stored in my schema (includes functions)

jdbc16Ora --

getProcedureColumns() -- parameter descriptions.

Class 08 13 08

Redundancy leads to anomalies which impact integrity

E.g. Rubio's DB

Table

empno ename pos sal bno baddr

110 jo mgr 40 1 india

120 mo mgr 35 2 japan

130 bo asst 20 1 india

140 jim asst 22 2 japan

150 kim mgr 40 3 france

160 kim asst 20 3 france

170 ralph asst 23 1 india

bno ? baddr baddr ? bno

Branches

bno baddr

1 india

2 japan

3 france

1. Insertion anomaly add an employee to table 1. 2. Deletion anomaly delete tim & kim. 3. Update anomaly each occurrence of France may need to be changed to Spain.

A,b,c ? function dependency if we know a, b and c, then we can determine d

a,b,c ? d a,b,c ? e

abbreviated by a,b,c ? d,e

Candidate key: minimum set of attributes, which uniquely determine a row in the database for any valid value in the table.

Salsbb02 grade role losal hisal g,r ? l, h a ? a

relation database, you are then in first normal form

 Big Bear Data Base

Rentals (ledger) client# cname prop# paddr startdate enddate rent owner# oname c12 jo p6 second 10:44, 14 August 2008 (CDT)~ 10:44, 14 August 2008 (CDT) [72.199.93.23](http://notewiki.gargantubig.com/index.php?title=User:72.199.93.23&action=edit&redlink=1) 052 FRED p8 3rd 10:44, 14 August 2008 (CDT)~ 10:44, 14 August 2008 (CDT) [72.199.93.23](http://notewiki.gargantubig.com/index.php?title=User:72.199.93.23&action=edit&redlink=1) 025 MOLLY c9 flo p6 second 10:44, 14 August 2008 (CDT)~ 10:44, 14 August 2008 (CDT) [72.199.93.23](http://notewiki.gargantubig.com/index.php?title=User:72.199.93.23&action=edit&redlink=1) 052 FRED p10 4th 10:44, 14 August 2008 (CDT)~ 10:44, 14 August 2008 (CDT) [72.199.93.23](http://notewiki.gargantubig.com/index.php?title=User:72.199.93.23&action=edit&redlink=1) 030 CY p8 3rd 10:44, 14 August 2008 (CDT)~ 10:44, 14 August 2008 (CDT) [72.199.93.23](http://notewiki.gargantubig.com/index.php?title=User:72.199.93.23&action=edit&redlink=1) 025 MOLLY

Constraints 1. No one rents 2 cabins at once 2. No one rents same cabin for 2 different weeks First normal form: every table entry is a scalar (can't put vector in there) (The above rentals database table is not really a true relational database table.) C are: kate it so//

Definition a1, a2, .. ak ? b is a partial dependency if b depends on a proper subset of {a1,.., ak} an attribute is prime if it belongs to some candidate key

Definition an attribute a is prime if is in so me candidate key. Aka: 2NF violation

b should not be in a candidate key (i.e. b should not be prime).

O# ? oname C# ? cname

p# 00> paddr, rent, o#, oname Clients client # cname c12 jo c9 flo

Proportion p# paddr rent o# oname p6 2nd [72.199.93.23](http://notewiki.gargantubig.com/index.php?title=User:72.199.93.23&action=edit&redlink=1) 052 FRED p8 3rd [72.199.93.23](http://notewiki.gargantubig.com/index.php?title=User:72.199.93.23&action=edit&redlink=1) 025 MOLLY p10 4th [72.199.93.23](http://notewiki.gargantubig.com/index.php?title=User:72.199.93.23&action=edit&redlink=1) 030 CY

Rentals c# p# start end

Transitive Dependency is A ? B ? C

^ ^ ^

| | |

cand non- non-prime key prime

Third normal form : no trans. dependencies

p# ? o# ? oname cand- ^ ^ key | | non-prime

what you get rid of when normalizing are the columns to the right of the arrow

Database Normalization

Will give us table and column names. Based on functional dependencies for table, we need to find 1) candidate keys 2) prime attributes 3) is it in second normal form? If not, make it so 4) is it in third normal form, if not make it so. 5) is it in 3.5 normal form? If not, make it so.

BCNF = Boyce-Codd normal form

T(D,O,N,T,C,R,Y) D,O ? N,T,C,R,Y C,R ? D D ? N (a) Candidate keys {D,O} {C,R,O} (b) Prime attributes {D,O,C,R}

1 NF Yes 2 NF No 3 NF BCNF

T2(D,N) T(D,O,T,C,R,Y) T3 (C,R,D) T(O,C,R,Y,T)

1 NF Yes 2 NF Yes 3 NF Yes BCNF

 A ? P A ? Q A ? P,Q

Grades (SID,Course#,Sem#,Grade) SID, C#, S# ? Grade

This is more typical of what would be on an exam:

Stuff(H,I,J,K,L,M,N,O) H,I ? J,K,L J ? M

 Store\_item(SKU, Promotion ID, Vendor, Style, Price) SKU, Promotion ID ? Vendor, Style, Price SKU ? Vendor, Style

Roo(t,h,i,s,e,x,a,m) t,i ? h,e,x,a t ? s t ? m a ? t x ? e

class 0818 2008

NF is Normal Form

Roo(t,h,i,s,e,x,a,m)

t,i ? h,e,x,a

t ? s

t ? m

a ? t

x ? e

(a) candidate keys

{t,i} {a,i}

(b) prime attributes

{t,i,a}

(c) 1NF -- Yes

2NF -- No

3NF -- No

BCNF --

put it in 2NF t-> s,m is a 2NF Violation

combine into biggest right side

Foo(\_t\_,s,m)

Ignore the word superkey unless youre asked to define it on an exam

Foo(\_t\_,s,m) Goo(\_x\_,e) Roo(t,h,i,x,a)

Foo(\_t\_,s,m) Goo(\_x\_,e)

Moo(\_a\_,t) Roo(h,\_i\_,x,\_a\_)

BCNF Compliant

Stuff table STUFF(\_H\_,\_I\_,J,K,L,M,N,O) H,I --> J,K,L

T1(J,M) T2(K,N) T3(L,O)

STUFF(\_H\_,\_I\_,J,K,L)

Shipping

Shipping(Shipname,Shiptype,VoyageID,Cargo,Port)

Shipname --> Shiptype

VoyageID --> Shipname, Cargo

Shipname,Date --> VoyageID,Port

Candidate Keys

{Shipname,date},{VoyageID,date}

Prime Attributes {Shipname,VoyageID,Date}

1NF --> yes

Ships (Shipname,Shiptype)

Voyages(VoyageID,Cargo)

Shipping(shipname,VoyageID,Port,Date)

2NF -->

3NF --> ok from 2nf fix

VIDSH(VoyageID,Shipname

Shipping(\_VoyageID\_,port,\_Date\_)

LOT of points wednesday on normalization 1,2,3NF BCNF Candidate key prime attribute superkey Be able to normalize

requirements analysis conceptual database design logical database design schema refinement -- normalization physical DB design what DBA does Application security design Refactoring (ie, change from soc to redid)

normalization cut up big into small tables denormalization make big table from little tables. joins can slow things down so denormalization can help

**employee table**

entity:Employee --properties: eid,ename,lot

entity in a box properties in circles joined with arrows primary keys underlined.

department -- \_did\_,dname,budget

(Since)

|

^

\_\_\_\_\_\_\_\_ / \ \_\_\_\_\_\_\_\_\_\_

[employee]==<works in >--[department]

\ /

v

^

\_\_\_\_\_\_\_\_ / \ \_\_\_\_\_\_\_\_\_\_

[employee]--< Manages > <==[department]

\ /

v

--> at most owe {key constraint bounded constraint

== at least one participation constraint

==> Exactly one

^

\_\_\_\_\_\_\_\_ / \ =============

[employee]--< policy > <==|[dependent ]|

\ / =============

v

|

( cost )

Heavy Box is a weak entity. entirely dependent

**Hierarchy**

[employee]

|

^

/ \

/ \

/ is a \

---------

/ \

/ \

\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

|hourly| |contract|

-------- ----------

what to do with subclasses? circle is a subclass of oval.

CREATE TABLE emp (eid PRIMARY KEY ename lot) CREATE TABLE dept (did PRIMARY KEY dname budget)

(Since)

|

^

\_\_\_\_\_\_\_\_ / \ \_\_\_\_\_\_\_\_\_\_

[employee]==<works in >--[department]

\ /

v

|

----------

|locaton |

----------

/ \

(\_address\_) (capacity)

CREATE TABLE works\_in(eid ... did... address... since...)

primary key, then main attributes

PRIMARY KEY(eid,did,address) FOREIGN KEY(eid)REFERENCES employee FOREIGN KEY(did) REFERENCES department FOREIGN KEY(address) REFERENCES location

(since)

|

^

\_\_\_\_\_\_\_\_ / \ \_\_\_\_\_\_\_\_\_\_

[employee]--< Manages > <--[department]

\ /

v

CREATE TABLE MANAGES (eid... did... since...) PRIMARY KEY (did)

(since)

|

^

\_\_\_\_\_\_\_\_ / \ \_\_\_\_\_\_\_\_\_\_

[employee]-->< Manages > <--[department]

\ /

v

2 at most ones. CREATE TABLE Manages (eid...did...since...) PRIMARY KEY (did) UNIQUE (eid)

^

\_\_\_\_\_\_\_\_ / \ \_\_\_\_\_\_\_\_\_\_

[employee]==<works in >==[department]

\ /

v

CREATE TABLE Works in(eid...NOT NULL did...since...) PRIMARY KEY (eid,did) FOREIGN KEY eid REFERENCES emp FOREIGN KEY (did)

test final quiz

cursors exceptions triggers 18 things on jdbc stuff done on blackboard er and normalization

canonical translation of relatonships translate picture to manages (translate entities) nothing on trying to create hard constraints

is a -- subclass weak entities why is it called weak? you cant get a primary key even with all columns

aggragation: relationship that points to another relationship --- dotted box