Entity Integrity constraint: Primary Key must be unique and not "NULL" GN UNKNOWN of original relation can be O a value of an existing primary key value of corresponding primary key in referenced relation, or O"NUL". highly repeated, thus the other outtrhutes in Department table are (W) TRUE It we put all the attributes of Department table in Employee table, Reterential integrity constraint - The value in toreign key column (i) TRUE we are wasting space. Because the values of attillante PNO are DB Mangement Midterm HW, 110705017 17至71年受好 also the same, which is not necessary to list them all. Foreign Key: The Key that reterence the primary key. Logic Layer Business Patabase resentation LAYER Octobase Database Management Server System Application Application Programs Server . Web Pages Clent | Web Interface GUI

Semant, constraints = Beyond the expressive power of model and must

specified and entered by application programs.

Servius Layer

Schema-based constraints: Express in schema by using-facilities provided by mode,

(b) Cx. (AR(State, RegHt, Sevial 196, Make, Mode), Kear)

Minimal Superkey: {Sevial 196, Make, Mode), Kear)

Minimal Superkey: {Sevec, Rg#; Sevially, Moke, Model, Kear)

AVG Hanction return nothing, (DVNT traction return 0.

(b) COUNT (Salary) accommunate each record To table thates not "NUL".

(c) COUNT (Althint Salary) calculate the number of different salary.

(d)

VPDATE COMPETITION

SET SCORE = 'A'

WHERE Sung-id IN (SELECT Song-id

FROM SONG

WHERE Sung-id IN (SELECT Song-id

FROM SONG

WHERE Sung-id IN (SELECT Student-number

FROM SONG

WHERE Sung-id IN (SELECT Student-number

FROM STUDENT

WHERE Sung-id IN (SELECT Student-number

FROM STUDENT

WHERE MAJON = "MIS")

SELECT S. NAME
FRON STUDENT AS S, COMPETITION AS C, SONB AS G
WHERE S. Sex = 'male'
AND S. Student\_number = C. Student\_number
AND S. Student\_number = C. Student\_number
AND C. Song\_Id = Gr. Song\_Id
AND C. Song\_Id = Gr. Song\_Id
AND G. Language = 'English'
AND G. Language = 'English'

(1)
SELECT E. FNAME, E. LNAME, COUNT(\*)
FROM PEPARTMENT AS D, EMPLOYEE AS E
(WHERE COUNT(\*)(FROM E WHERE E. PNO = P. PNUMBER)>3
(2)
SELECT R. PNAME, E. FNAME, E. LNAME
FROM PROJECT AS P, EMPLOYEE AS E, PEPARTMENT AS P
WHERE P. PNAME = 'M.LS'

SELECT E. FNAME, E. LNAME, COUNT(\*)
FROM EMPLOYEE AS E, DEPENDENT AS P, WORKS-ON AS W
WHERE E. SSN= P. ESSN 25 NULL

AND F. DNUM = D. DNUMBER AND E. DNO = D. DNUMBER AND (SELECT PISTINCT W. PNO

FROM E, D, W WHERE E.SN=W.ESSN) HAVING (OUNT(\*) > 2

FROM EMPLOYEE AS SON M. SUPERSSN = S. SSN  INNER JOIN EMPLOYEE AS SON M. SUPERSSN = S. SSN  WHERE M. SSN IN (SELECT MGRSSN)  FROM DEPARTMENT  WHERE PNUMBER NOT IN (SELECT  DNUM FROM PROJECT) AND M. SSN NOT IN (SELECT

0

WHERE W. PNO = P. PNUMBER) > (SELECT MAX (MISCOUNT)
FROM (SELECT COUNT (\*) AS MISCOUNT
FROM WORK-ON AS WL, PROJECT AS PZ
WHERE WZ. PNO = PZ. PNUMBER
ANP PZ. DNUM IN (SELECT DNUMBER FROM

WHIERE (SELECT COUNTC\*) FROM WORKS\_ON AS W

FROM PROJECT AS P

SELECT P. PNAME

FROM MORKS-ON)

PEPARTMENT WHERE DNAME = "MIS" GROUPBY P. DNUM

SELECT DESTINCT E. FNAME, E. LAMME  FROM EMPLOYEE AS E, PROJELT AS P  FROM WORKSLON AS W  JOLN EMPLOYEE AS ELON WHERE NO POPE AS ELON WHERE W. PNO = P. PNUMBER,  AND EL. KNAME = 'John'  AND EL. KNAME = 'John'  AND EL. KNAME = 'Switch')  EXCEPT CSELECT (*)  FROM WORKSLON AS WL  WHERE WL. PNO = PNUMBER  AND WL. ESSN = E. ESSN)  WITH RECURSIVE SUMER Employees (Supsish) Enpsish)  CO  WITH RECURSIVE SUMER Employees (Supsish) ELECT  YOURON Super Employees AS ELECT  SELECT SE SUPSIN E SSN  FROM Super Employee AS E ON S. Empsish  JOLN Super Employee AS E ON S. Empsish  SELECT E. FNAME, E. LAMME, SEL SNS SEL ENPSIN  JOLN Super Employees AS ELON E. SSN = SEL ENPSIN  SELECT E. FNAME, E. LAMME, SEL SNS = SEL ENPSIN  JOLN Super Employees AS SEL ON SEL EMPSIN = SEL ENPSIN	
SELECT DISTINCT E.FNAME, E.LNAME  FROM EMPLOYEE AS E, PROJELT AS P WHERE NOT EXISTS ((SELECTA*)  FROM WORKS_ON AS W  JOIN EMPLOYEE AS E-SSN WHERE W.PNO = P. PNUMBER, AND EL.LNAME = 'John' AND EL.LNAME = 'John' AND EL.LNAME = 'Swith')  EXCEPT (SELECT (*)  FROM WORKS_ON AS WIL WHERE WIL PNO = PNUMBER, AND WITH RECURSIVE SUPERIEDPHYEES ((Supsisin))  (B) WITH RECURSIVE SUPERIEDPHYEES ((Supsisin))  (C) WITH RECURSIVE SUPERIEDPHYEES ((Supsisin))  (C) WITH RECURSIVE SUPPOSING  (SELECT Supsisin)  FROM Superiemphyees AS SEL MANE, SEL TNAME FROM Superiemphyees AS SEL ON S. EmpSin  SELECT E. FNAME, E.LNAME, SEL TNAME, SEL LNAME FROM EMPLOYEES AS SEL ON S. EmpSin  SELECT S. FNAME, E.LNAME, SEL EMPSIN  SELECT E. FNAME, E.LNAME, SEL FNAME, SEL LNAME FROM EMPLOYEES AS SEL ON SEL EMPSIN = SEN EMPSIN  JOLN Superiemphyees AS SEL ON SEL EMPSIN = SEN EMPSIN	(1)
FROM EMPLOYEE AS E, PROJECT AS P WHERE NOTEXISSUSELECTE  WHERE NOTEXISSUSELECTE  ON W.ESSIN = ED. SSIN WHERE W. PNO = P. PNUMBER.  AND ED. LNAME = 'John' AND ED. LNAME = 'John' AND ED. LNAME = 'Smith')  EXCEPT CSELECT(*)  WHERE W. PNO = PNUMBER. AND WALFE W. PNO = FNUMBER. AND WALESIN = ESSIN)  WITH RECURSIVE SUMEYEMPLOYGES (ScySSIN) EmpSSIN   ESSIN)  CSELECT Supersing SSIN   E. SSIN FROM Super Employees AS SELON E. SSIN  SELECT E. SAMPE, E. LNAME, SEL. FNAME; SEL. LNAME FROM EMPLOYEES AS SELON E. SSIN SELECT E. FNAME; SEL. LNAME FROM EMPLOYEES AS SELON E. SSIN SELECT E. FNAME; SEL. EMPSSIN JOIN Super Employees AS SELON E. SSIN SELECT  SELECT E. FNAME, E. LNAME, SEL. FNAME, SEL. EMPSSIN JOIN Super Employees AS SELON E. SSIN SEL. EmpSSIN JOIN Super Employees AS SELON E. SSIN SEL. EmpSSIN JOIN Super Employees AS SELON E. SSIN SEL. EmpSSIN	SELECT DISTINCT E. FNAME, E. LNAME
WHERE NOTEXISTS ((SELECTR)  WHERE WORKLON AS W  JOIN EMPLOYEE AS EL  ON W.ESSN = EL. SSN  WHERE W. PNO = P. PNUMBER  AND EL. LNAME = 'John'  AND EL. LNAME = 'Smith')  EXCEPT (SELECT R')  FROM WORKS ON AS WL  WHERE WL. PNO = PNUMBER  AND WL. ESSN = E. ESSN)  (SELECT SIMPSSN)  ENDSN / Empssn  FROM Super Employees (Supssn)  SELECT SE. Supssn  JOIN Super Employees AS EL ON S. Empssn  JOIN Super Employees AS SEL ON E. SSN = SSN    SELECT E. FNAME, E. LNAME SEL: FNAME  FROM ENRIOYEE AS E  JOIN Super Employees AS SEL ON E. SSN = SEL Empssn  SOLN Super Employees AS SEL ON E. SEN  JOIN Super Employees AS SEL ON E. SSN = SEL Empssn  SELECT E. FNAME, E. LNAME  JOIN Super Employees AS SEL ON E. SIN = SEL Empssn  JOIN Super Employees AS SEL ON SEL Empssn = SEL Empssn  SOLN Super Employees AS SEL ON SEL Empssn = SEL Empssn	FROM EMPLOYEE ASE, PROJECT ASP
727777	WHERE NOT EXISTS ((SELECTR)
	FROM WORKS_ON AS W
	JOIN EMPLOYEE AS EL
727777	NSS. 53 = NSSSN = E2. SSN
2 Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z	WHERE W. PNO = P. PNUMBER
TO T	AND EZ. FNAME = JOhn!
	AND EL. LNAME = 'SMITH')
7277777	EXCEPT (SELECT (*)
1 N T N S T	FROM WORKS-ON AS WIL
T D S H D D S L	WHERE WZ. PNO = PNUMBER
MITH RECURSIVE Super Employees (SupSSN) EmpSSN) AS  (SELECT SupSSN) EmpSSN.  FROM Super Employees AS SE  JOLN Super Employees AS SE  JOLN Employee AS E ON SENEMPSSN = E.SSN)  SELECT E. FNAME, E. LNAME, SEL. FNAME, SEL. LNAME  FROM Super Employees AS SEI ON E.SSN = SELLARME  JOLN Super Employees AS SEI ON E.SSN = SELLARME  TRON Super Employees AS SEI ON E.SSN = SELLARME  JOLN Super Employees AS SEI ON E.SSN = SELLARME	88
FROM SupEmp  UNION ALL SELECT SE, SupSSN, E, SSN  FROM SuperEmployees AS SE  JOIN SupEmp AS S ON SE, EmpSSN = S, SupSSN  JOIN Employee AS E ON S, EmpSSN = E, SSN)  SELECT E, FNAME, E, LNAME, SEL, FNAME, SEL, LNAME  TRON EMPLOYEE AS E  JOIN SuperEmployees AS SELON E, SSN = SEL, EmpSSN  JOIN SuperEmployees AS SELON E, SSN = SEL, EmpSSN	WITH RECURSIVE SUPEREMPROYEES (SUPSSN, EmpSSN) AS
FROM Sup Emp  UNION ALL  SELECT SE, SupSSN, E, SSN  FROM SuperEmployees AS SE  JOLN SupEmp AS S ON SE, EmpSSN = S, SupSSN  JOLN Employee AS E ON S, EmpSSN = E, SSN)  SELECT E, FNAME, E, LNAME, SEL, FNAME, SEL, LNAME  TROW EMPLOYEE AS E  JOLN Super Employees AS SEL ON E, SSN = SEL, EmpSSN  JOLN Super Employees AS SEL ON E, SSN = SEL, EmpSSN	(SELECT Sup SSN, EmpSSN
SELECT SE, SupSSN, E, SSN FROM SuperEmployees AS SE JOLN SupEmp AS S ON SE, EmpSSN = S, SupSSN JOLN Employee AS E ON S, EmpSSN = E, SSN) SELECT E, FNAME, E, LNAME, SEL, FNAME, SEL, LNAME FROM EMPLOYEE AS E JOLN SuperEmployees AS SEI ON E, SSN = SEI, EmpSSN JOLN SuperEmployees AS SEI ON E, SSN = SEI, EmpSSN	FROM Sup Emp
SELECT SE, SupSSN, E, SSN FROM SuperEmployees AS SE JOLN SupEmp AS SON SE, EmpSSN = S, SupSSN JOLN Employee AS E ON S, EmpSSN = E, SSN) SELECT E, FNAME, E, LNAME, SEL, FNAME, SEL, LNAME TROW EMPLOYEE AS E JOLN SuperEmployees AS SEI ON E, SSN = SEI, EmpSSN JOLN SuperEmployees AS SEI ON E, SSN = SEI, EmpSSN	UNION ALL
FROM SuperEmployees AS SE  JOLN Supemp AS SON SE. EmpSSN = S. SupSSN  JOLN Employee AS E ON S. EmpSSN = E. SSN)  SELECT E. FNAME, E. LNAME, SEL. FNAME, SEL. LNAME  FROM EMPLOYEE AS E  JOLN SuperEmployees AS SELON E.SSN = SEL. EmpSSN  JOLN SuperEmployees AS SELON E.SSN = SEL. EmpSSN	■ SELECT SE, SupSSN, E, SSN
JULIN SUPERMP AS SON SEIEMPSON = J.SupSSN  JOIN EMPLOYEE AS E ON S. EMPSON = E. SSN)  SELECT E. FNAME, E. LNAME, SELIFNAME, SELILNAME  FROM EMPLOYEE AS E  JOIN Super Employees AS SEI ON E.SSN = SEI. EmpSSN	FROM Super Employees AS SE
SELECT E. FNAME, E. LNAME, SEL. FNAME, SEZ. LNAME  FRON EMPLOYEE AS E  JOIN Super Employees AS SEI ON E.SSN = SEI. EmpSSN  JOIN Super Employees AS SEI ON E.SSN = SEI. EmpSSN	JULIN SUPERMP AS SON SE, EMPSSN = S. SWYSSN
FRON EMPLOYEE AS E  JOIN Super Employees AS SEI ON E.SSN = SEI. EmpSSN  JOIN Super Employees AS SEI ON E.SSN = SEI. EmpSSN	DOLN Employee AS E ON S. EmpSSN = E.SSN)
JOIN Super Employees AS SEI ON E.SSN = SEI. EmpSSN	- SELECT E. FNAME, E. LNAME, SEL, FNAME, SEL, LNAME FROM EMPOYET NOT
JOEN Super Employees AS SEZ ON SELL Emp SSN = SE, GAZ STA	JOIN Super Employees AS SELON E SON - SEL F. SCO.
	JOIN Super Employees AS SE2 ON SEl. Emp SSN = (F) G. (CK)
	NECONC 17C - COMM

(SELECT SupSSN  (SELECT SupSSN  FROM SupEnp WHERE EmpsSN = E.SSN  GROUP BY SupSSN  (SROUP BY SupSSN  (A) Array (\$-POST ['dept-name'], \$-POST ['proj-loc'])  (B) \$-POST ['dept-name']  (B) \$alliesult as \$r  (A) \$alliesult as \$r  (B) \$r[0]  (B) \$ r[1]  (C) \$ radvising as \$student = 7 \$advisor)  (D) \$cence (\$advising as \$student = 7 \$advisor)  (E) \$cence \$cencent. (bro);  (E) \$cence (\$advising as \$student = 7 \$advisor)  (E) \$cence \$cencent. (bro);  (E) \$cence \$cencent. (bro);  (E) \$cence \$cencent. (bro);  (E) \$cence \$cencent. (bro);  (E) \$cencent. (		)(°(')°(')	Tim = > (Parre (Visor)
[ERE E. SEX = 'Male' AN  (SELECT SUPSSN  FROM SupeEmp  WITERE EmpsSN = F  GROUP BY SUPSN = F  SPOST ['dept-name']  1	P SE1, SupSSN= =. SSN	17, 5-805TL'pr	n'=> 'John', ' tudent => \$ ac advised by '. \$ a
FRE E.SE (SELECT   FROM   WITER   GROUP   S-POST [1-1-4]   S-POST [1-1-4-4]   S-POST [1-1-4-4]   S-POST [1-1-4-4-4]   S-POST [1-1-4-4-4-4]   S-POST [1-1-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4	X=1Mu e1 AN Sup Emp Sup Emp = = Emp SSN = 7 BY Sup SSN	st t dept-name ot-name] proj-locij as \$ r	Jack ) j Jack ) j advising as \$s advising as \$s
0 2 2 8 4 9 9	HERE E.SE,  (SELECT FROM WHERE GROUP	(1) array (\$-80 (2) \$-8057 ['deg (3) [\$-8057 ['deg (4) \$allresult (5) \$ V[0] (6) \$ V[0]	a) \$ advising = > 1 Mary = > 1 b) foreach (\$ if echo \$ct