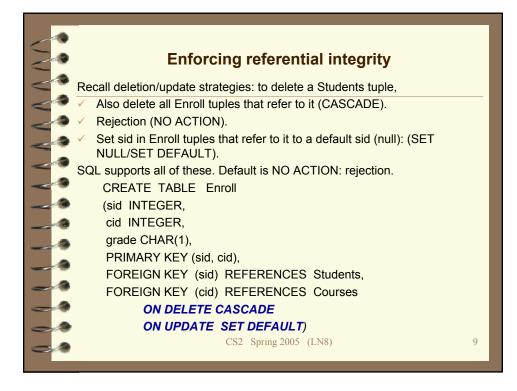
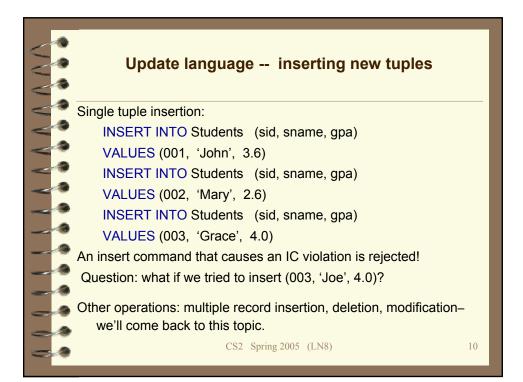
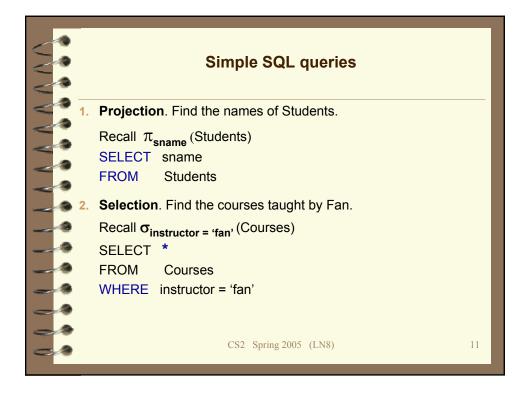
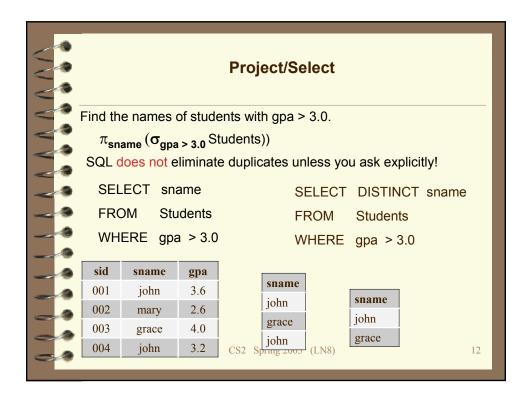


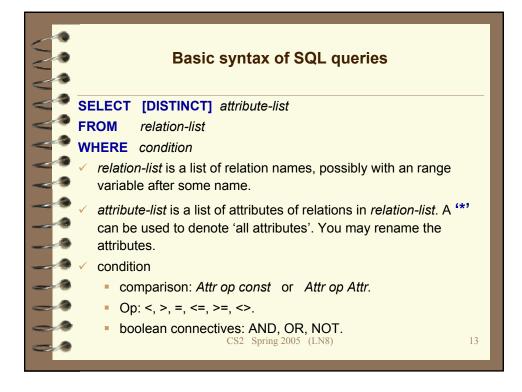
	Null values	
	Attribute values in a tuple are sometimes <i>unknown</i> or <i>inapplicable</i> (e.g., no spouse's name for a single). These are treated as a special value: null	
	Keys cannot have null values (but foreign keys can) Three-valued logic:	
	 Comparison operations (e.g., >) e.g., 3 < null unknown 	
	 Logic connectives (e.g., AND, OR, NOT) 	
	o false AND unknown? False o true OR unknown? True	
	o false OR unknown? unknown	
C 1	CS2 Spring 2005 (LN8)	8

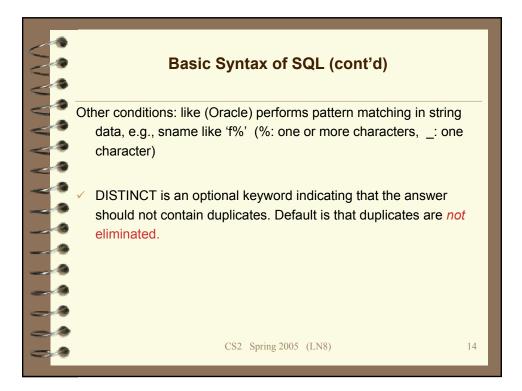


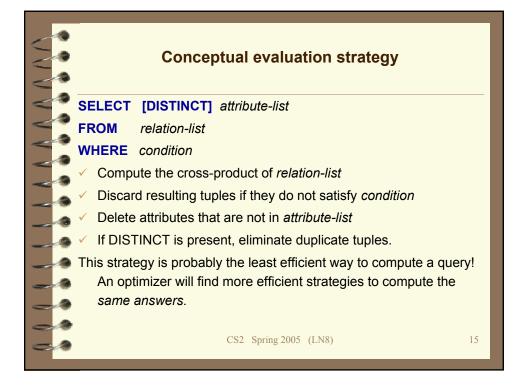




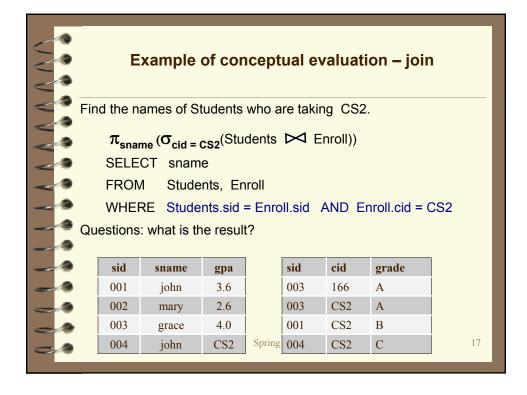




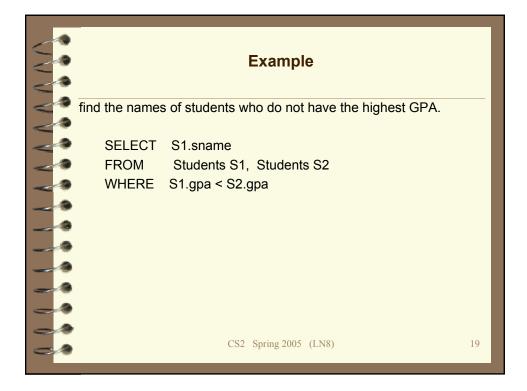


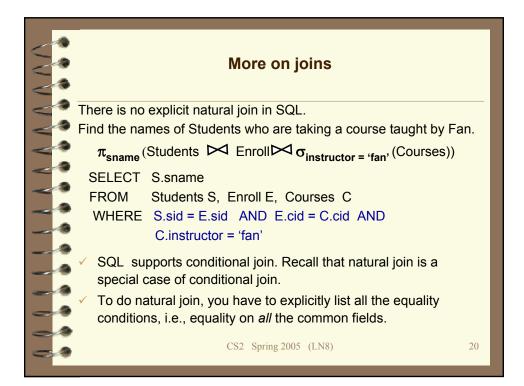


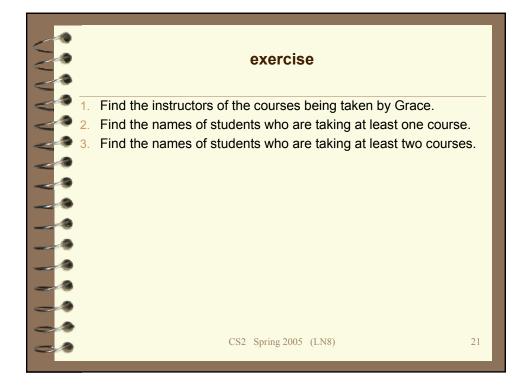
1 2	Exam	ple of	со	nceptual e	valuati	on -	- pro	oduc	t
	SEL	ECT *							
FROM Students, Enroll									
sid	sname	ma	1	Students.sid	sname	gpa	sid	cid	grade
001	john	e gpa 3.6		001	john	3.6	003	166	А
002	5			002	mary	2.6	003	166	А
003	grace	4.0		003	grace	4.0	003	166	А
				001	john	3.6	003	CS	А
sid	cid	grade		002	mary	2.6	003	$\frac{2}{CS}$	А
003	166	А			mary			2	A
003	CS2	А		003	grace	4.0	003	CS	А
				CS2 Spring 2005	(LN8)			-	1



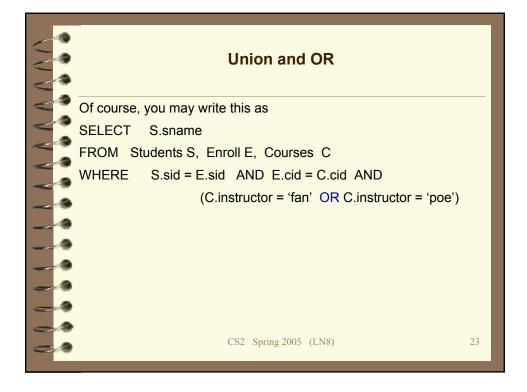
		_
10		_
20	A note on range variables	
1		_
<	Find the names of Students who are taking CS2.	
<3	It is a bit awkward to write Students.sid	_
_3	SELECT sname	
	FROM Students, Enroll	
20	WHERE Students.sid = Enroll.sid AND Enroll.cid = CS2	
	We can rewrite it using range variables:	
	SELECT S.sname	
	FROM Students S, Enroll E	
	WHERE S.sid = E.sid AND E.cid = CS2	
	 Really needed only if the same relation appears twice in the FROM 	_
	clause.	
-3	 It is good style, however, to use range variables all the time! 	
	CS2 Spring 2005 (LN8)	18
0		

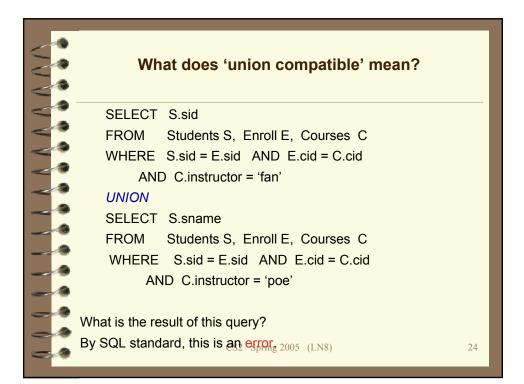




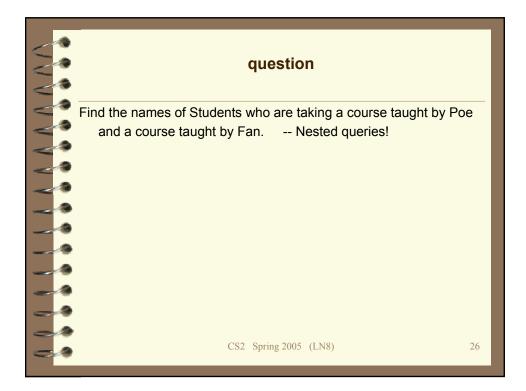


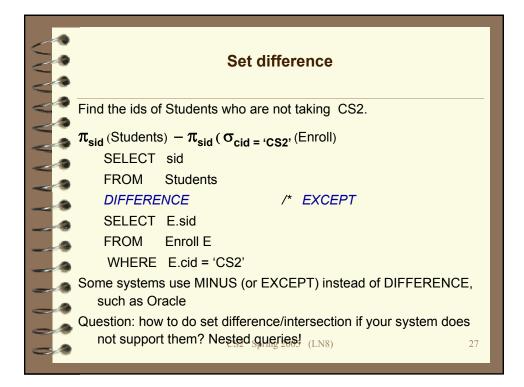
A A A	Union
V	 Find the names of Students who are taking a course taught by Poe or Fan.
2	SELECT S.sname
_	FROM Students S, Enroll E, Courses C
~	WHERE S.sid = E.sid AND E.cid = C.cid
	AND C.instructor = 'fan'
	UNION /* UNION ALL reserves duplicates
	SELECT S.sname
	FROM Students S, Enroll E, Courses C
	WHERE S.sid = E.sid AND E.cid = C.cid AND
	C.instructor = 'poe'
0 0	CS2 Spring 2005 (LN8) 22

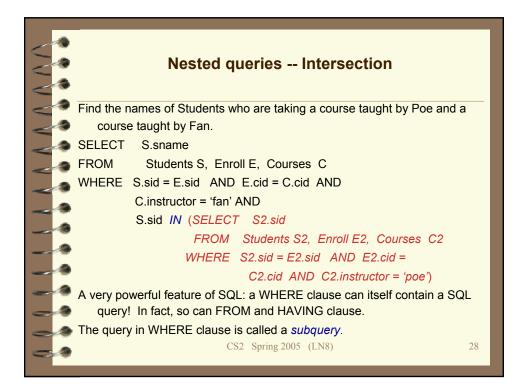


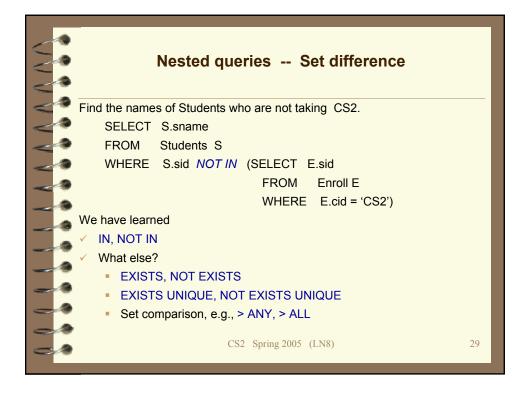


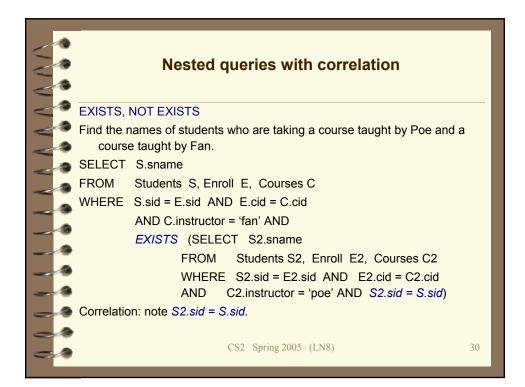
	Intersection
<.	Find the ids of Students who are taking a course taught by Poe and a course taught by Fan.
<	π_{sid} (Enroll) $\sigma_{instructor = 'poe'}$ (Courses)) \cap
~	π_{sid} (Enroll $\sigma_{instructor = 'fan'}$ (Courses)) SELECT S.sid
	FROM Enroll E, Courses C
	WHERE E.cid = C.cid AND C.instructor = 'fan' INTERSECT
	SELECT S.sid
	FROM Enroll E, Courses C
-	WHERE E.cid = C.cid AND C.instructor = 'poe'
-3	CS2 Spring 2005 (LN8) 25

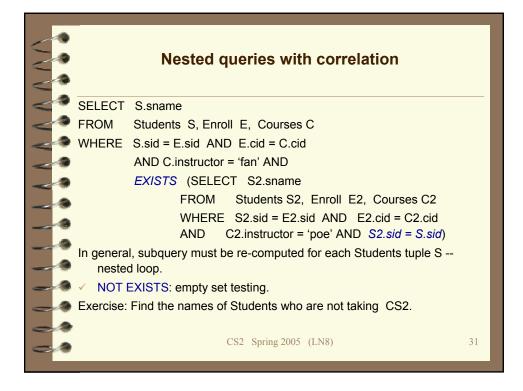






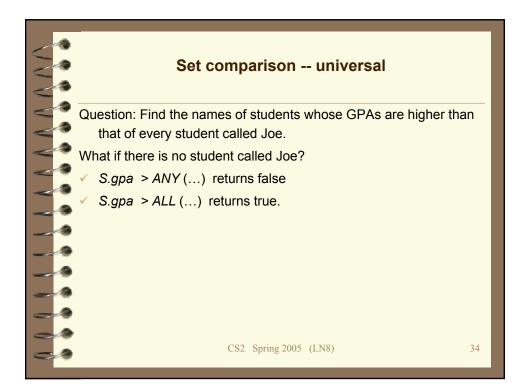


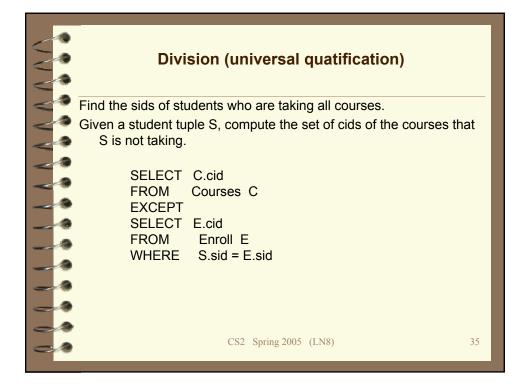




More on nested queries
EXISTS UNIQUE, NOT EXISTS UNIQUE: checking for duplicate tuples. True only if no two tuples appear more than once in the answer to the subquery.
 Find the names of students who either do not take CS2, or don't take 166. SELECT S.sname FROM Students S
WHERE EXISTS UNIQUE (SELECT DISTINCT E.sid FROM Enroll E WHERE S.sid = E.sid AND E.cid = CS2
UNION ALL SELECT DISTINCT E.sid FROM Enroll E CS2 Spring 2005 (LN8) 32
WHERE S.sid = E.sid AND E.cid = 166)

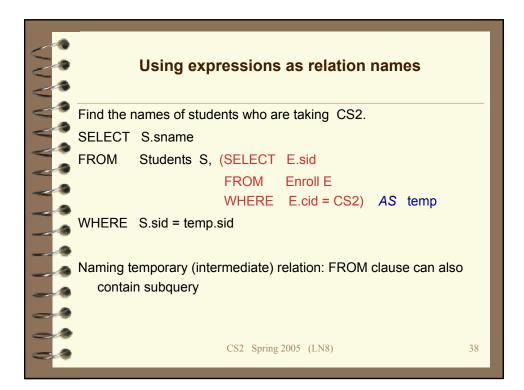
~~~~	Set comparison operations	
</th <th><i>op ANY, op ALL,</i> where op: &gt;, &lt;, =, &lt;&gt;, &gt;=, &lt;=.</th> <th></th>	<i>op ANY, op ALL,</i> where op: >, <, =, <>, >=, <=.	
<	ANY: there exists some (existential). ALL: for all (every), universal.	
20	Find the names of students whose GPAs are higher than that of some student called Joe.	
	SELECT S.sname	
	FROM Students S	
	WHERE S.gpa > ANY (SELECT S2.gpa	
	FROM Students S2	
-0	WHERE S2.sname = 'Joe')	
-1		
	CS2 Spring 2005 (LN8) 33	3

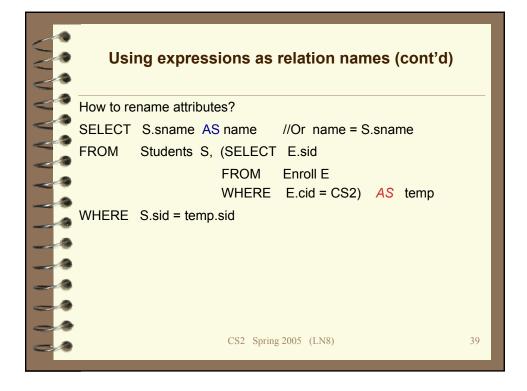


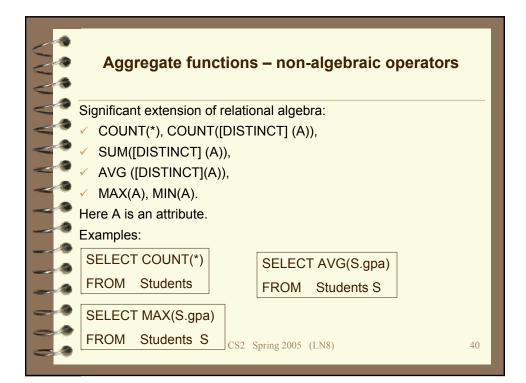


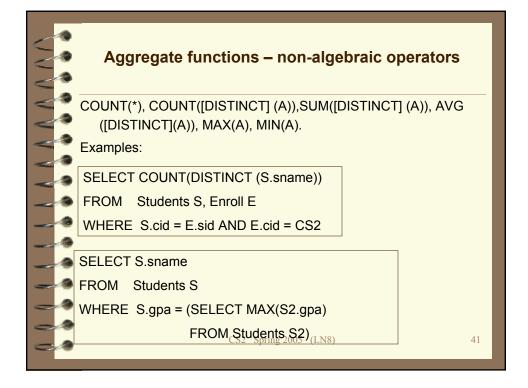
	Division (cont'd)	
	S is put in the answer if and only if the set is empty! SELECT S.sname FROM Students S	
	WHERE NOT EXISTS (SELECT C.cid FROM Courses C EXCEPT SELECT E.cid	
	FROM Enroll E WHERE S.sid = E.sid)	
9 9 9	CS2 Spring 2005 (LN8) 36	

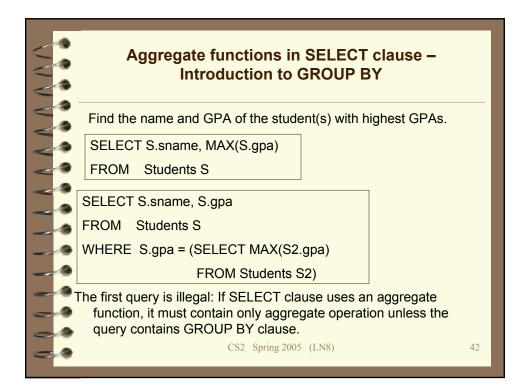
More on division	
Find the names of students who are taking all courses taught by Fan.	
SELECT S.sname	
FROM Students S	
WHERE NOT EXISTS (SELECT C.cid	
FROM Courses C	
WHERE C.instructor = 'fan'	
EXCEPT	
SELECT E.cid	
FROM Enroll E	
WHERE S.sid = E.sid)	
Exercise: rewrite the query without using DIFFERENCE.	
CS2 Spring 2005 (LN8)	37

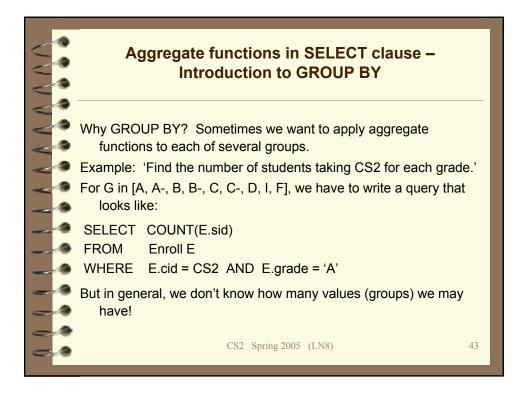




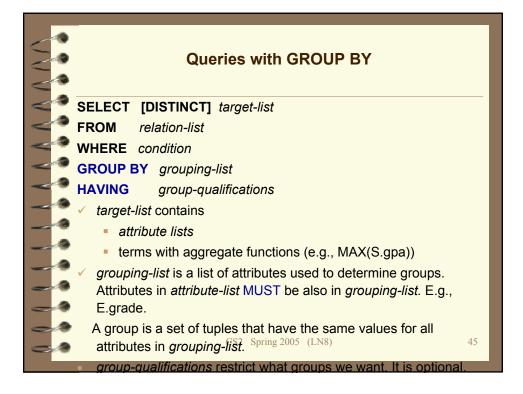


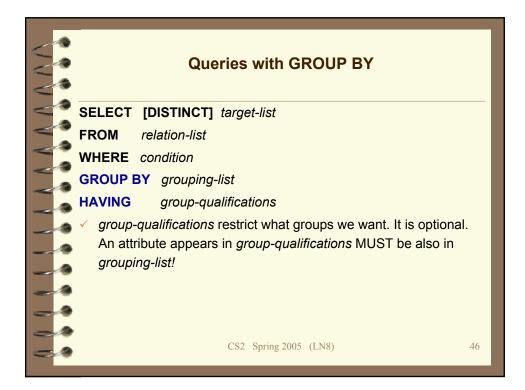


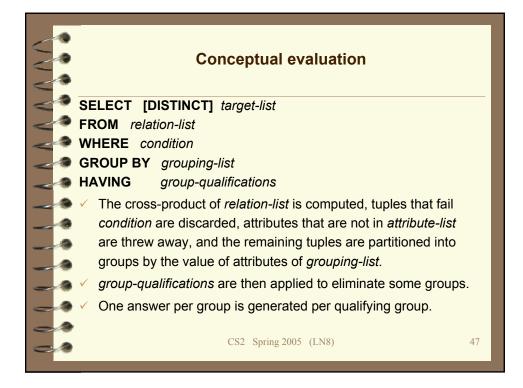


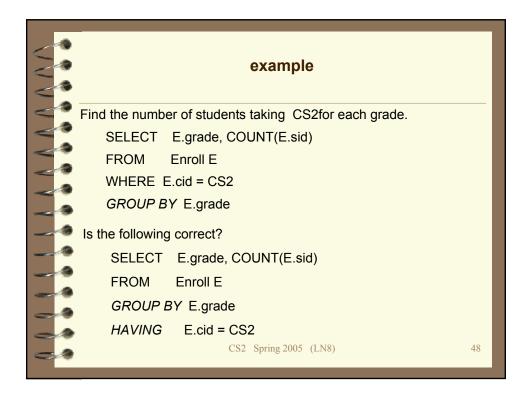


<	
20	Group by
20	
</th <th>For each grade, find the number of CS2 students receiving that grade.</th>	For each grade, find the number of CS2 students receiving that grade.
< 3	SELECT E.grade, COUNT(E.sid)
<	FROM Enroll E
</th <th>WHERE E.cid = CS2</th>	WHERE E.cid = CS2
~?	GROUP BY E.grade
	For each grade higher than 'F', find the number of CS2students receiving
	that grade.
	SELECT E.grade, COUNT(E.sid)
3	FROM Enroll E
- 0	WHERE E.cid = CS2
-3	GROUP BY E.grade
-	<i>Having</i> E.grade > 'F'
-	CS2 Spring 2005 (LN8) 44









	More on GROUP BY	
	✓ Is the following correct?	
~	SELECT E.sid, E.grade, COUNT(E.sid) FROM Enroll E	
<	WHERE E.cid = CS2 GROUP BY E.grade	
	Find the average GPA of students for each course with credit > 2 taught by each instructor.	
3 3 3	SELECTC.instructor, C.credit, AVG(S.gpa)FROMStudents S, Enroll E, Courses CWHERES.sid = E.sid ANDE.cid = C.cidGROUP BYC.instructor, C.credit	
~* ~*	HAVING C.credit > 2 CS2 Spring 2005 (LN8)	49

