SQL TABLE					
CREATE	TABLE [SCHEMA]	TABLE NAME	AS (SELECT….)		
INSERT INTO	TABLE_NAME	[(COLUMN_NAME,……)]			
			" A | | ' ' | | B"		
	SELECT	[DISTINCT | UNIQUE]	(*, COLUMN_NAME [ARITHMATIC OPERATOR] [AS ALIAS],…])	FUNCTION(COLUMN_NAME)	SINGLE-ROW SUBQUERY
		FROM [DUAL]	TABLENAME1, TABLENAME2, ….	TABLENAME1 [CROSS APPLY (SELECT statement)]/ [RIGHT|LEFT|FULL] OUTER/CROSS/NATURAL JOIN TABLENAME2	MULTIPLE-COLUMN SUBQUERY
				ON (CONDITION)/USING(COLUMN_NAME)	
		WHERE	CONDITION (COMPARISON [LOGICAL OPERATOR|ROWNUM])/	FUNCTION(COLUMN_NAME)	SINGLE-ROW SUBQUERY, IN/ANY/ALL/EXISTS MULTIPLE-ROW SUBQUERY, IN MULTIPLE-COLUMN SUBQUERY
		FOR UPDATE (If desired to update); Next claused will not be used then.			
		GROUP BY	EXPRESSION	GROUPING SETS(COLUMN_NAME1, ….), CUBE(COLUMN_NAME1,…), ROLLUP(COLUMN_NAME1,…)	
		HAVING	GROUP_CONDITION		SINGLE-ROW SUBQUERY, IN/ANY/ALL/EXISTS MULTIPLE-ROW SUBQUERY, IN MULTIPLE-COLUMN SUBQUERY
		ORDER BY	COLUMN_NAME|COLUMN_POSITION|EXPRESSION [ASC|DESC][NULLS FIRST|NULLS LAST]		
		FETCH FIRST N ROWS/PERCENT ROWS ONLY			
	UNION/UNION ALL/INTRSECT/MINUS SELECT ….				
	CREATE	TABLE [SCHEMA]	TABLE NAME		
		(COLUMN_NAME	DATA_TYPE [DEFAULT] VALUE INVISIBLE/PRIMARY/REFERENCE COLUMN/UNIQUE/CHECK(CONDITION)		
		CONSTRAINT CONSTRAINT_NAME	CONSTRAINT_KEYWORD (COLUMN_NAME)	)	
		ORGANIZED INDEX			
	ALTER	TABLE	TABLE_NAME		
		ADD/MODIFY/DROP/SET UNUSED/ DISABLE/ENABLE	COLUMN/CONSTRAINT(DEFAULT ON NULL value)		
	TRUNCATE	TABLE	TABLE NAME		
	RENAME	TABLE	TABLE NAME AS		
	DROP	TABLE			
	FLASHBACK	TABLE	TABLE_NAME TO BEFORE DROP		
	DROP	TABLE	TABLE NAME PURGE		
	PURGE	TABLE			
	DESC	TABLE			
	INSERT INTO	TABLE_NAME [(COLUMN_NAME,……)]			
		VALUES (datavalue)			
	UPDATE	TABLE_NAME			
		SET COLUMN_NAME = VALUE			
		WHERE CONDITION			
	DELETE FROM	TABLE_NAME			
		WHERE CONDITION			
	ROLLBACK				
	ROLLBACK TO	HOW MANY REDO			
	COMMIT				
	LOCK TABLE	TABLE_NAME			
	MERGE INTO	TABLE_NAME			
		USING			
		ON			
		WHEN MATCHED THEN			
		UPDATE SET			
		WHERE CONDITION			
		WHEN NOT MATCHED THEN			
		INSERT			
		VALUES (datavalue)			
		WHERE CONDITION			
	CREATE	SEQUENCE	SEQUENCENAME		
		INCREMENT BY value			
		START WITH value			
		MAXVALUE value| NOMAXVALUE			
		MINVALUE value| NOMINVALUE			
		CYCLE|NOCYCLE			
		ORDER|NOORDER			
		CACHE value| NOCACHE			
	ALTER	SEQUENCE			
		INCEREMENT BY value			
	DROP	SEQUENCE	SEQUENCENAME		
	CREATE	INDEX/ BITMAP INDEX/	INDEXNAME		
		ON TABLENAME(COLUMNNAME)			
	DROP	INDEX			
	CREATE	SYNONYM	SYNONYMNAME		
		FOR	OBJECT		
	DROP	SYNONYM			
	CREATE OR REPLACE	[FORCE|NOFORCE] VIEW/ MATERIALIZED VIEW	VIEWNAME (COLUMNNAME,…)		
		AS SELECT statement			
		WITH CHECK OPTION [CONSTRAINT constraintname]			
		WITH READ ONLY			
	DROP	VIEW			

Element	Description	Example
MONTH	Name of the month spelled out and padded with blank spaces to a total width of nine characters	APRIL
MON	Three-letter abbreviation for the name of the month	APR
MM	Two-digit numeric value for the month	04
RM	Roman numeral representing the month	IV
D	Numeric value for the day of the week	Wednesday = 4
DD	Numeric value for the day of the month	28
DDD	Numeric value for the day of the year	December 31 = 365
DAY	Name of the day of the week, padded with blank spaces to a length of nine characters	WEDNESDAY
DY	Three-letter abbreviation for the day of the week	WED
YYYY	Displays the four-digit numeric value of the year	2009
YYY or YY or Y	The last three, two, or single digits of the year	2009 = 009; 2009 = 09; 2009 = 9
YEAR	Spelled-out version of the year	TWO THOUSAND NINE
B.C. or A.D.	Value indicating B.C. or A.D.	2009 A.D.

Time Elements		
SS	Seconds	Value between 0–59
SSSS	Seconds past midnight	Value between 0–86399
MI	Minutes	Value between 0–59
HH or HH12	Hours	Value between 1–12
HH24	Hours	Value between 0–23
A.M. or P.M.	Value indicating morning or evening hours	A.M. (before noon) or P.M. (after noon)

Number Elements		
9	Indicates width of display with a series of 9s, but insignificant leading zeros are not displayed	99999
0	Displays insignificant leading zeros	0009999
$	Displays a floating dollar sign	$99999
.	Indicates number of decimals to display	999.99
,	Displays a comma in the position indicated	9,999

```
--Determine which books cost less than the average cost
-- of other books in the same category..
SELECT a.title, b.category, a.cost
FROM books a, (SELECT category, AVG(cost) averagecost
FROM books
GROUP BY category) b
WHERE a.category = b.category
AND a.cost < b.averagecost;
--Determine which orders had a higher total amount due than order 1008
SELECT ORDER#, SUM(QUANTITY*PAIDEACH)
FROM ORDERITEMS
HAVING SUM(QUANTITY*PAIDEACH)>ALL(SELECT SUM(QUANTITY*PAIDEACH)
FROM ORDERITEMS
WHERE ORDER# = 1008
GROUP BY ORDER#)
GROUP BY ORDER#;
--Determine which author or authors wrote the books most frequently
--purchased by customers of JustLee Books.
SELECT lname, fname
FROM bookauthor
JOIN author USING(authorid)
WHERE isbn IN(SELECT isbn
FROM orderitems
GROUP BY isbn
HAVING SUM(quantity) =(SELECT MAX(COUNT(*))
FROM orderitems
--List the title of all books in the same category as
--books previously purchased by customer 1007.
--Don't include books this customer has already purchased.
SELECT title
FROM books
WHERE category IN
(SELECT DISTINCT category
FROM books JOIN orderitems USING(isbn)
JOIN orders USING(order#)
WHERE customer# = 1007)
AND isbn NOT IN
(SELECT isbn
FROM orders JOIN orderitems USING(order#)
WHERE customer# = 1007);
--Determine the number of different customers who have
--placed an order for books written or
--cowritten by James Austin.
SELECT COUNT(DISTINCT customer#)
FROM orders JOIN orderitems USING(order#)
WHERE isbn IN
(SELECT isbn
FROM orderitems JOIN bookauthor USING(isbn)
JOIN author USING(authorid)
WHERE lname= 'AUSTIN'
AND fname = 'JAMES');
--Problem 03
CREATE FORCE VIEW HOMEWORK13
AS SELECT COL1, COL2 FROM FIRSTATTEMPT;
```

red Query Language file