id	difficulty	stars	description	predicate	hints
S1	straightforward	1	Get all suppliers.	Supplier with number `sid` is named `name`, has status `status` and is located in the city named `city`	"Get all..." and "Get … (unqualified)" means "with all available attributes".
S2	straightforward	1	Get the number and name of suppliers.	Supplier with number `sid` is named `name`	"Get x, y, and z …" means "x, y and z attributes only".
S3	straightforward	1	Get all but the status of suppliers.	(There exists a status such that) supplier with number `sid` is named `name` and is located in city named `city`	"Get all but x, y and z" means "all available attributes except x, y and z"
S4	straightforward	1	Get the city names where at least one supplier is located.	(There exists a supplier, say s, … such that) s is located in city named `city`	In accordance with relational theory, "the city names" means "no duplicates".
S5	straightforward	1	Get suppliers with status in the range 15 to 25 inclusive.	[Supplier with number `sid` is named `name`, has status `status`, is located in city named `city`] and his status is in the range 15 to 25 inclusive	
S6	straightforward	1	Get suppliers located in London or Athens.	[…] and that city is named London or is named Athens	[…] means "the common predicate for the relation variable at hand"
S7	straightforward	1	Get suppliers not located in London nor in Athens.	[…] and that city is not named London nor Athens	
S8	straightforward	1	Get supplier numbers for suppliers located in London.	Supplier with number `sid` is located in London	
E1	easy	2	Get supplier numbers for suppliers who supply part P1.	Supplier with number `sid` supplies the part whose number is P1	
E2	easy	2	Get how many shipments there are.	The total count of shipments is `c`	The result is a single scalar. To keep the automation of corrections simple enough, we want a relation with one single tuple and one single attribute named `c`.
E3	easy	2	Get how many parts are supplied by supplier S1.	Supplier whose number is S1 supplies `c` parts	By convention, an unknown supplier supplies no part at all, accounting for zero.
E4	easy	2	Get suppliers who supply at least one part.	Supplier with number `sid` is named `name`, has status `status`, is located in the city named `city` and supplies at least one part (i.e. there exists a part number such that…)	
E5	easy	2	Get suppliers who supply no part at all.	Supplier with number `sid` [is named …] and supplies no part at all (i.e. there does not exist…)	negation, exists, forall
E6	easy	2	Get shipments together with the name of the supplier (sname) and the name of the part (pname)	Supplier with number `sid` is named `sname` and ships the part with number `pid`, named `pname` in `qty` quantity.	
M1	medium	4	Get the `total` quantity of parts supplied by supplier S1.	Supplier whose number is S1 supplies parts for a total quantity of `total`	By convention, an unknown supplier supplies no part at all, for a total quantity of zero. SQL may seem to lack consistency… Does Tutorial D do?

M2	medium	4	Get part numbers for parts supplied by at least one supplier in Paris.	The exists at least one supplier located in Paris who supplies the part whose number is `pid`	
M3	medium	4	Get city names for cities in which at least two suppliers are located.	There exists at least two suppliers located in the city named `city`	At least means ">="
M4	medium	4	Get supplier numbers for suppliers whose city is first in the alphabetic list of such cities.	Supplier with number `sid` is located in the city named `city` and that city name appears first in the alphabetic list of supplier cities.	
M5	medium	4	Get the supplier numbers and the `total` quantity of supplied parts for each of them.	Supplier with number `sid` supplies parts for a total quantity of `total`	Including those who do not seem to supply any part
M6	medium	4	Get the number(s) of lightest part(s). In case of ex-aequo, get all parts with that smallest weight.	Part with number `pid` has the smallest known weight among all parts	
D1	difficult	8	Get part numbers for parts not supplied by any supplier in Paris.	No supplier in Paris supplies the part with number `pid`	
D2	difficult	8	Get numbers of suppliers who supply all parts.	Supplier with number `sid` supplies all parts.	What if no parts at all?
D3	difficult	8	Get supplier for suppliers with a status strictly lower than of supplier S1.	Supplier with number `sid` is named […] and has a status strictly lower than supplier with number S1	What if supplier S1 does not exist?
D4	difficult	8	Get supplier-number/part-number pairs such that the indicated supplier does not supply the indicated part.	Supplier with number `sid` does not supply part with number `pid`	
D5	difficult	8	Get every known city together with the count of parts located in that city (`ps`), the count of suppliers located there (`ss`), and the total shipped quantity by the latter (`total`).	In city named `city`, there are `ps` located parts, `ss` located suppliers and a total quantity of `total` shipped parts by those suppliers.	**ALL** known cities
D6	difficult	8	Get the heaviest part. In case of ex-aequo, select the one with the smallest part number.	Part number `pid` is the first in the alphabetic list of (part numbers of) heaviest parts	
H1	hard	16	Get every city name where at least one part is located, together with the average supplier status in that city (`avg_status`). Do not include cities where no supplier is located.	At least one part is located in the the city named `city` and that city has located suppliers for an average status of `avg_status`	
H2	hard	16	Get part numbers for parts supplied by all suppliers in London.	Part with number `pid` is supplied by all suppliers located in London	

H3	hard	16	Get all pairs of supplier numbers, say sx and sy, such that sx and sy supply exactly the same set of parts each.	Supplier with number `sx` supplies the same parts as supplier with number `sy`.	"the same parts" means the same set of part numbers, that is, abstracting from quantities. We require here supplier numbers to be distinct (i.e. sx <> sy) and consider ordered pairs, that is, (sx=S1,sy=S2) is different than (sx=S2,sy=S1).
H4	hard	16	Get all pairs of part numbers (as `p` and `q`) such that some supplier supplies both of the indicated parts.	There exists a supplier supplying both part with number `p` and part with number `q`	We require here part numbers to be distinct (i.e. p <> q) and consider ordered pairs, that is, (p=P1,q=P2) is different than (p=P2,q=P1).
H5	hard	16	Get supplier numbers for suppliers supplying at least as many parts as the three "best" suppliers (in terms of supplied quantities), together with their respective total quantity.	With a total quantity of supplied parts `total`, supplier with number `sid` is among those who supply the three largest total quantities of parts.	The result does not necessarily contain three tuples; it might contain more (why?), or less (why?).