SQL to dplyr cheat sheet

| **SQL** | **R dplyr** |
| --- | --- |
| **SELECT** *var1, var2, ...* FROM *table* | select(*table, var1, var2, ...*)  -or-  *table* %>% select(*var1, var2, ...)* |
| SELECT **DISTINCT** *var1, var2, ...*  FROM *table* | distinct(*table, var1, var2, ...)*  -or-  *table* %>%distinct*(var1, var2, ...)*  distinct(*table*)  # uses all variables in table |
| SELECT *var1* **AS** *renamed\_var, var2, ...*  FROM *table* | select(*table, renamed\_var = var1, var2, ...*)  -or-  *table* %>% select(*renamed\_var = var1, var2, ...)*  # drops unnamed variables  rename(table, *renamed\_var = var1*)  -or-  *table %>%* rename(*renamed\_var = var1*)  # keeps unnamed variables  mutate(table, renamed\_var = var1)  -or-  table %>% mutate(renamed\_var = var1)  # keeps unnamed variables |
| SELECT *var1* + 10 **AS** *calc*, *var2, ...* | mutate(*table, calc = var1* + 10)  -or-  *table* %>% mutate (*calc = var1* + 10)  # keeps unnamed variables  transmute(table, *calc = var1* + 10*, var2 = var2*)  -or-  *table %>%* transmute (*calc = var1* + 10*, var2 = var2*)  # drops unnamed variables |
| SELECT \* FROM *table*  **WHERE** *condition1* AND *condition2*  SELECT \* FROM *table*  **WHERE** *condition1* OR *condition2* | filter(*table, condition1 & condition2)*  -or-  filter(*table, condition1, condition2)*  -or-  *table* %>% filter(*condition1, condition2)*  filter(*table, condition1 | condition2*)  -or-  table %>% filter(*condition1 | condition2*) |
| SELECT \* FROM *table*  **ORDER BY** *var1, var2* DESC | arrange(*table, var1, desc(var2), ...*)  -or-  *table* %>% arrange(*var1, desc(var2), ...)* |
| SELECT \* FROM *table1*  **INNER JOIN** *table2*  BY *t1.var1 = t2.var1*  AND *t1.var2 = t2.var2*  *--* variables have same names  BY *t1.var1 = t2.var3*  AND *t1.var2 = t2.var4*  -- variables have different names | inner\_join(*table1, table2, by =* c(“*var1*”*,* “*var2*”)  -or-  *table1* %>% inner\_join(*table2, by =* c(“*var1*”*, “var2*”)  # variables have same names  inner\_join(*table1, table2,*  *by =* c(“*var1” = “var3”, “var2” = “var4”*))  -or-  *table1* %>% inner\_join(*table2,*  *by =* c(“*var1” = “var3”, “var2” = “var4”*))  # variables have different names |
| INNER JOIN  LEFT JOIN  RIGHT JOIN  FULL JOIN  WHERE EXISTS  WHERE NOT EXISTS  INTERSECT  UNION  EXCEPT | inner\_join()  left\_join()  right\_join()  full\_join()  semi\_join()  anti\_join()  intersect()  union()  setdiff() |
| SELECT *var1*,  COUNT(\*) AS var1\_num,  SUM(*var2*) AS *var2\_sum*  FROM *table*  **GROUP BY** *var1* | summarise(  group\_by(*table, var1*),  *var1\_num = n(),*  *var2\_sum =* sum(*var2*))  -or-  *table %>%*  group\_by(*var1*) %>%  summarise(*var1\_num* = n(),  *var2\_sum =* sum(*var2*)) |
| SELECT \* FROM (  SELECT \*, **RANK() OVER** (PARTITION BY *var1, var2* ORDER BY *var3, var4* DESC) AS rank1  FROM *table*  )  WHERE rank1 = 1 | filter(mutate(group\_by(*table, var1, var2*), *rank1* = *min\_rank(var3,* desc(*var4*))), rank1 = 1)  -or-  *table* %>%  group\_by(*var1, var2*) %>%  mutate(*rank1* = *min\_*rank*(var3,* desc(*var4*))) %>%  filter(rank1 = 1)  -or-  *table* %>%  group\_by(*var1, var2*) %>%  filter(min\_rank*(var3,* desc(*var4*)) = 1) |
| **CASE WHEN** *condition1* THEN *result1*  **ELSE** *result2* **END**  **CASE WHEN** *condition1* THEN *result1*  WHEN *condition2* THEN *result2*  WHEN *condition3* THEN *result3*  ELSE*result4*  END | if\_else(*condition1, result1, result2)*  case\_when(  *condition1* ~ *result1,*  *condition2* ~ *result2,*  *condition3* ~ *result3,*  TRUE~ *result4*) |
| *var1* = *var2*  *var1* **AND** *var2*  *var1* **OR** *var2*  **NOT** *var1*  *var1* **IS NULL**  *var1* **IS NOT NULL**  *var1* **IN** (‘*opt1*’, ‘*opt2*’, ‘*opt3*’) | *var1* == *var2*  *var1* & *var2*  *var1* | *var2*  !*var1*  is.na(*var1*)  !is.na(*var1*)  *var1* %in% c(‘*opt1*’, ‘*opt2*’, ‘*opt3*’) |
| CEIL(*var1*)  FLOOR(*var1*)  NCHAR(*var1*)  CONCAT(*var1, var2*), ||  MIN(*var1*)  MAX(*var1*)  LENGTH(*var1*)  LOWER(*var1*)  UPPER(*var1*)  COUNT(*\**)  COUNT(DISTINCT *var1*)  DENSE\_RANK() OVER (ORDER BY *var1*)  RANK() OVER (ORDER BY *var1*)  ROW\_NUMBER() OVER (ORDER BY *var1*)  *var1* BETWEEN *val1* AND *val2*  NVL(*var1, val1*)  COALESCE(*var1, var2, val1*) | ceiling(*var1*)  floor(*var1*)  length(*var1*)  paste0(*var1, var2*)  max(*var1)* –or sometimes- pmin(*var1*)  min(*var1)* –or sometimes- pmax(*var1*)  str\_length(*var1*)  str\_to\_lower(*var1*)  str\_to\_upper(*var1*)  n()  n\_distinct(*var1*)  dense\_rank(*var1*)  min\_rank(*var1*)  row\_number(*var1*)  between(*var1, val1, val2*)  replace\_na(*var1, val1*)  coalesce(*var1, var2, val1*) |