A `CHECK` condition evaluating to `UNKNOWN` passes — it only fails if the condition is `FALSE`.

#2c — a "greater than or equal to 0" constraint like `No_of_copies >= 0` can be implemented as a `CHECK` constraint or via `UNSIGNED` on the column. `UNSIGNED` is a MySQL extension so `CHECK` is more portable. (An advantage of `UNSIGNED` is that the sign bit can be used for values, so it doubles the range of valid values.)

#2g — `LIKE '___-___-____'` is sufficient to match the condition given in the problem. To require the `X` characters to be digits means using MySQL's `REGEX` (a non-standard extension) instead of `LIKE` — worth knowing about, but not covered in class or expected knowledge.

#2h — the possibility of `Returned_date` being `NULL` has the potential to throw a wrench in things.

```
CHECK (Due_date >= Date_out AND Returned_date >= Date_out))
```

works because `TRUE AND UNKNOWN` is `UNKNOWN` and `CHECK` passes `UNKNOWN` and `FALSE AND UNKNOWN` is `FALSE`. (Separate `CHECK`s for each condition also works.) It's clearer, though, to explicitly handle the `Returned_date IS NULL` case.

#2l — addressing the actions for all three affected tables (`BOOK_AUTHORS`, `BOOK_COPIES`, and `BOOK_LOANS`) is clearest, but the answer is still technically correct if you only listed the things different from the schema given in the homework.

#2n — `CHECK (CURDATE() > Date_out)` is correct in spirit, but `CHECK` constraints can't use non-fixed values (like `CURDATE`). A `CHECK` constraint holds for all of the data in the database all of the time — it's not just a gatekeeper check when something is inserted. With `CURDATE`, the check might pass when the loan is insert but then become false down the road.

#3 — Files created in Windows need `LINES TERMINATED BY '\r\n'` but that's not safe for non-Windows files.

#4c — A composite foreign key addresses the main point. A `UNIQUE` constraint on the referred-to columns is also needed because of a technicality — a foreign key can only reference one row in the referred-to table. Normally foreign keys refer to primary keys; since it doesn't in this case, a `UNIQUE` constraint on the referred-to columns is needed to ensure only one row is referenced.

#4d — `DECIMAL` is incomplete since it doesn't cover the "tenth of an hour" part. e.g. `DECIMAL(4,1)`